Exploration of Factors Affecting Intention to Use Software Testing Tool in Jinan, Shandong, China with Quantitative Survey Method

Hua Xiao, Chia Yean Lim*

*School of Computer Sciences, Universiti Sains Malaysia, Minden, Penang, Malaysia. *Email: cylim@usm.my*

The adoption of the right software testing tools for software quality checking is important to ensure the comprehensiveness of software accuracy before the software is launched and used by the intended users. However, there is a persistent gap in understanding the factors that influence the intention to use software testing tools among software testers. This research was conducted with quantitative survey analysis to explore whether the quality factors of functional stability, reliability, usability, security, and portability have positive and significant relationships with the intention to use software testing tools, among the 306 software testers from multi-field companies. All the hypothesis testing for the above-mentioned relationships were accepted. In subsequent multiple regression tests, three factors namely reliability, usability, and portability are significant and could be used to construct the theoretical framework for software testing tools adoption. The findings of this research could help the developers of software testing tools to create more reliable, high usability, and portable testing tools, to increase the software adoption rate among the software testers.

Keywords: Intention to Use, ISO 25010 model, Software Testing Tools, Survey Analysis.

INTRODUCTION

Observation of the execution of a software system through software testing is an essential step within the software development life cycle (SDLC). As software applications grow increasingly complex within the evolving technology landscape, the importance of robust and effective software testing has become more pronounced [1]. Testing can be divided into five core phrases that start from Analysis, followed by Planning and Preparation, and Execution to Closure [2]. The objective of software testing is to guarantee the provision of a high-quality and faultless product that fulfills the requirements of the user. The significance of software testing lies in the potential consequences of software failures, which can vary from minor inconveniences to critical malfunctions within systems. There are diverse variations of testing methodologies, such as unit testing, integration testing, and system testing [3].

The increasing complexity of modern software systems has led to the development of different methods for testing software. Traditional methods like manual testing, where human testers execute test cases without automated tools are still crucial for applying subjective judgment in the testing process. However, the increasing need for speed and efficiency has driven the widespread adoption of automated testing tools and frameworks. Automated testing tools enable the rapid execution of repetitive test cases, offering enhanced efficiency and accuracy compared to manual testing [4].

The importance of selecting and effectively utilizing software testing tools has grown significantly with organizations relying more heavily on software for critical business processes [5]. Despite the availability of numerous testing tools, there is a gap in understanding the factors that influence the behavioral intention of software professionals to adopt and use these tools [6]. This study addresses the issue of the absence of a comprehensive framework that systematically integrates the various factors influencing the behavioral intention to use software testing tools. The objective of this study is to bridge this gap by proposing a novel software tools' acceptance framework that enhances user adoption and satisfaction with the utilization of software testing tools. The research seeks to understand what influences software testers to use software testing tools to create a framework. It aims to find out which factors positively relate to the intention of incorporating these tools in framework development.

LITERATURE REVIEW

Overview of software testing tools

It is important to prioritize the quality and dependability of software applications in the constantly changing field of software development. Software testing plays a significant role in this process as it systematically evaluates the functionality and performance of software to identify and resolve errors. One of the main benefits of using software testing tools is the acceleration of the testing process. Software testing tools make test case execution more efficient by automating repetitive and time-consuming testing tasks [7]. Another significant impact of software testing tools is the enhancement of test coverage [8]. Finally, cost-effectiveness emerges as a significant aspect of the impact of software testing tools on software development [9].

Technologies are designed specifically to enhance the testing process, automate repetitive tasks, and offer a comprehensive analysis of the software's behavior. Software testing tools typically fall into three main categories: load testing tools, test management tools, and functional testing tools. Table 1 illustrates the categories, examples, and applications of various software testing tools.

Table 1 Categories of software testing tools

Categories	Examples	Uses
Load testing tools	WebLoad, JMeter [10]	Assess the performance of an application under various loads
Test management tools	Jira	Maintain comprehensive information regarding testing

		methodology
Functional testing tools	Selenium, JUnit	Designed to automate and facilitate the process of functional testing

On the other hand, non-functional testing tools are designed to assess the non-functional facets of an application, covering areas including usability, performance, scalability, reliability, security, compatibility, and other related aspects [11]. Different types of tools for non-functional testing can be grouped under an umbrella term. An example of this is performance testing, which involves evaluating the system's capacity and efficiency [12]. Non-functional testing evaluates the overall quality of the product rather than focusing on its features.

Landscape of Software Testing Tools

As technology advances rapidly, software testing tools are becoming more advanced, varied, and integrated into the overall development lifecycle. A significant trend in today’s software testing tool landscape was the extensive embrace of test automation. Organizations increasingly acknowledge the advantages of automating repetitive and labor-intensive testing tasks to enhance efficiency [5]. Moreover, artificial intelligence (AI) and machine learning (ML) have entered the realm of software testing, propelled by the prevailing trend of automation. AI-powered testing tools use sophisticated algorithms to automate the generation of API tests and facilitate visual validation in automation testing [13]. Cloud computing has a substantial impact on the landscape of software testing tools. Cloud-based testing solutions that offer scalability, flexibility, and cost-effectiveness services enable organizations to leverage resources based on their personal needs [14]. There is an increasing demand for testing tools tailored to these environments. These tools aim to streamline the testing process for applications developed using visual interfaces and minimal coding. A testing tool with low-code capabilities empowers business users and citizen developers to ensure the functionality and performance of their applications without the need for significant technical expertise [15].

Comprehensive Review of Relevant Studies

This section explores two key theoretical frameworks essential for comprehending the adoption of technology and user behavior for using software testing tools. These frameworks are the Technology Acceptance Model (TAM) and the Unified Theory of Acceptance and Use of Technology (UTAUT).

Technology Acceptance Model (TAM)

Figure 1 illustrates an overview of the Technology Acceptance Model (TAM). The Technology Acceptance TAM is extensively employed in the field of information systems and technology management. Its primary purpose is to understand and predict user acceptance of novel technologies [16]. Davis [17] first proposed the Technology Acceptance Model in 1989. TAM underscores the importance of understanding the psychological factors that influence user acceptance and adoption of technology. Perceived usefulness and perceived ease of use are core components of TAM.

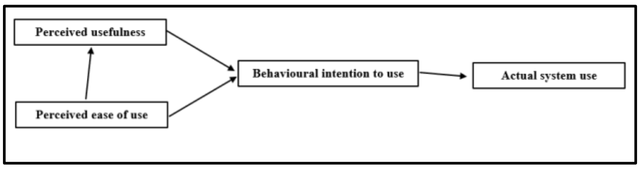


Figure 1 Overview of TAM model [16]

Unified Theory of Acceptance and Use of Technology (UTAUT)

Figure 2 presents an overview of the Unified Theory of Acceptance and Use of Technology (UTAUT), a well-recognized model devised by Venkatesh et al. [18] to explain and predict user acceptance and usage behaviors toward technology. The four primary factors along with the additional three moderating factors are elaborated and outlined in Table 2. Additional factors and moderating variables are integrated into the UTAUT model. It acknowledges the influence of individual differences on the acceptance of technology. This makes it better suited to capture the complexities of technology acceptance in diverse user populations.

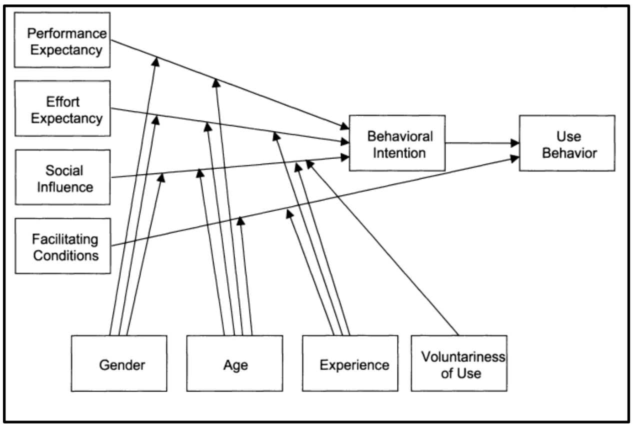


Figure 2 Overview of the UTAUT model [18]

Table 2 Variables in the UTAUT model

Types of variables	Variables	Description
Core	Performance Expectancy	Degree of believing technology will help them achieve specific goals.
	Effort Expectancy	Degree of believing using a particular technology will be free of effort.
	Social Influence	The influence exerted by social entities to use certain technology.
	Facilitating Conditions	Degree of believing resources and support systems facilitate the use of a particular technology.
Moderating	Gender	Influence of gender on technology adoption.
	Age	Influence of age groups to adopt certain technology
	Experience	Influence of previous experience to adopt certain technology

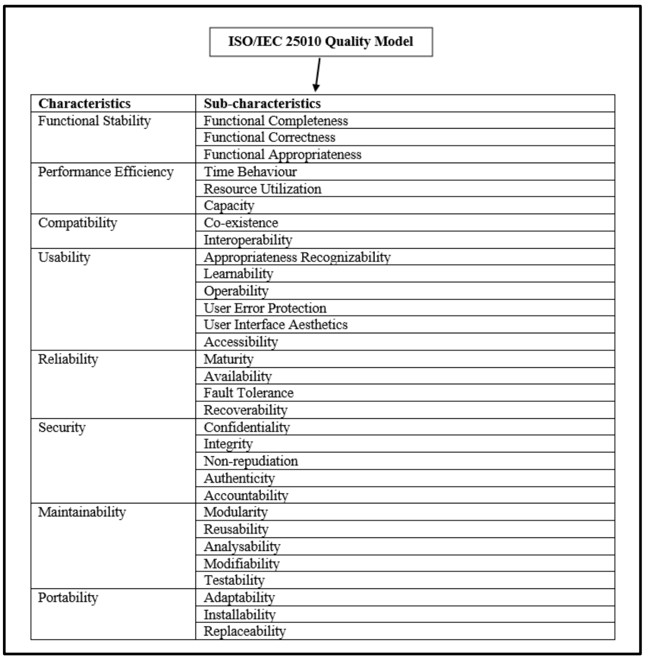
Table 3 presents a thorough summary of various studies conducted to explore the acceptance and utilization of technology across diverse contexts, by adopting the TAM or UTAUT model. Each row in the table describes a sampling approach, notable findings, authors, and the utilized model. The studies showed that the variables from the TAM or UTAUT model had significant relationships with user intention to use or acceptance behavior for a type of application system or tools.

Table 3 Overview of relevant study

Sampling	Findings	Author	Model
500 university lecturers and managers in a Jordanian university	Perceived usefulness and perceived ease of use significantly influence the acceptance of the ERP system	[19]	TAM
340 users of the Mobile Library Application (MLA)	There was significant impact of perceived usefulness and perceived ease of use on the intention to use MLA. System and habits are also influential factors	[20]	
120 students at Himachal Pradesh University	Performance expectancy, effort expectancy, attitude towards using, and facilitating conditions are highlighted as crucial factors in the research	[21]	UTAUT
880 student responses in Kuwait using a questionnaire survey	Performance expectancy, effort expectancy, and peer influence are key in determining students' behavioral intentions that ultimately influence the use of e-government services	[22]	

An In-depth Analysis of ISO Models and Their Components

In addition to the variables and factors from the popular technology acceptance models of TAM and UTAUT, this research also attempted to explore the significant relationships of the software quality variables from the ISO/IEC25010 with software testers' intention to use a software testing tool. ISO/IEC 25010 serves as an international standard for assessing the quality of software and systems [19]. Figure 3 presents a comprehensive outline of the model, including its characteristics and sub-characteristics. These characteristics offer a thorough perspective on the quality of a software product, ensuring it fulfills the requirements and expectations of stakeholders by addressing functionality, performance, compatibility, usability, reliability, security, maintainability, and portability.



ISO/IEC 25010 Quality Model

Characteristics	Sub-characteristics
Functional Stability	Functional Completeness
	Functional Correctness
	Functional Appropriateness
Performance Efficiency	Time Behaviour
	Resource Utilization
	Capacity
Compatibility	Co-existence
	Interoperability
Usability	Appropriateness Recognizability
	Learnability
	Operability
	User Error Protection
	User Interface Aesthetics
	Accessibility
Reliability	Maturity
	Availability
	Fault Tolerance
	Recoverability
Security	Confidentiality
	Integrity
	Non-repudiation
	Authenticity
	Accountability
Maintainability	Modularity
	Reusability
	Analysability
	Modifiability
	Testability
Portability	Adaptability
	Installability
	Replaceability

Figure 3 Overview of ISO/IEC 25010 model

A noteworthy example is the proposal by Aguirre et al. [23] to extend the concept of user satisfaction in e-learning environments by integrating the ISO/IEC 25010 standard in their study. This demonstrated the versatility of the ISO/IEC 25010 model beyond its initial application, showcasing its usefulness across various contexts. Another study by Fadhel et al. [24] aimed to devise a new theoretical framework for assessing system quality, incorporating both TAM and ISO/IEC 25010, emphasizing factors directly linked to user satisfaction. The findings indicated that ISO/IEC 25010 can serve as a reliable measure of success in systems quality engineering.

RESEARCH METHODOLOGY

The study examines the relationship between software quality metrics and the intention to use and it is structured in way of independent variables and dependent variables depicted in Table 4.

Table 4 Factor derivation source within the framework

Factor	Source	Types
Functional Stability	ISO/ IEC 25010 from Estdale and Georgiadu [25]	Independent variable
Reliability		
Usability		
Security		
Maintainability	Song and Sohn [26]	
Portability	Sidik and Syafar [27]	
Behavioral Intention	UTAUT [18]	Dependent variable

The factors of Functional Stability, Reliability, Usability, and Security are defined based on the guidelines provided in the ISO/IEC 25010 standard by referring to the study from Estdale and Georgiadu [25]. Maintainability is another independent factor derived from the findings of Song and Sohn [26], indicating its significant impact on the adoption of cloud computing across various groups, except for software as a service. On top of that, the Portability factor is derived from the research conducted by Sidik and Syafar [27], which highlighted the significant influence of quality of service on students' intention to use mobile learning. This influence is explained by indicators such as reliability, real-time, accessibility, portability, and interoperability that students believe will facilitate their learning outcomes.

Furthermore, Behavioral Intention is classified as a dependent factor in the study. It is drawn upon the extended version of the Technology Acceptance Model (TAM) known as the Unified Theory of Acceptance and Use of Technology (UTAUT) model proposed by Venkatesh et al. [18]. The focus of the factors is to investigate the intention to use software testing tools with an emphasis on software quality metrics. Some factors derived from the software quality metrics are related to use. For instance, security is a component of certain intention-to-use frameworks [28].

Figure 4 displays the hypotheses derived from the research framework. The hypotheses used to test the proposed research theoretical framework are as follows:

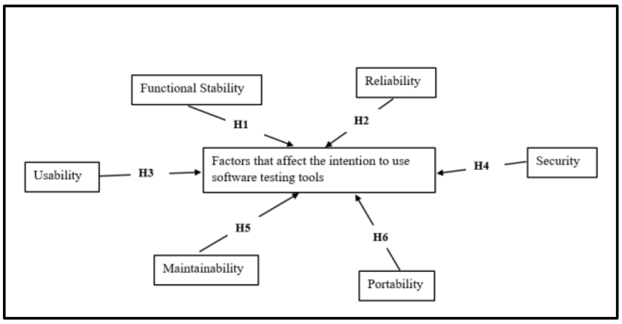


Figure 4 Theoretical framework of research

H1- Functional Stability is positive and significantly related to the behavioral intention to use software testing tool.

H2- Reliability is positive and significantly related to the behavioral intention to use software testing tool.

H3- Usability is positive and significantly related to the behavioral intention to use software testing tool.

H4- Security is positive and significant related to the behavioral intention to use software testing tool.

H5- Maintainability is positive and significantly related to the behavioral intention to use software testing tool.

H6- Portability is positive and significantly related to the behavioral intention to use software testing tool.

Figure 5 presents an overview of the study process in the form of a flowchart. It consists of six core phrases and certain core phrases can be further divided into multiple sub-phrases.

Table 5 offers a summary of the survey study population details, including the types of tests conducted, the target population, the location, and the inclusion criteria. The survey study was chosen as the primary research method due to its ability to directly gather input from individuals.

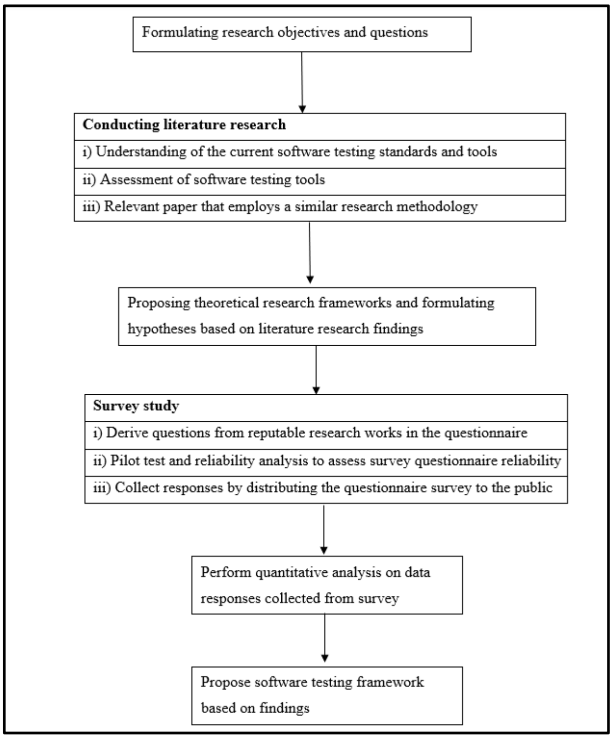

Formulating research objectives and questions
Conducting literature research
i) Understanding of the current software testing standards and tools
ii) Assessment of software testing tools
iii) Relevant paper that employs a similar research methodology
Proposing theoretical research frameworks and formulating hypotheses based on literature research findings
Survey study
i) Derive questions from reputable research works in the questionnaire
ii) Pilot test and reliability analysis to assess survey questionnaire reliability
iii) Collect responses by distributing the questionnaire survey to the public
Perform quantitative analysis on data responses collected from survey
Propose software testing framework based on findings

Figure 5 Flowchart of study

Table 5 Overview of the study population

Types of tests	Target population	Location	Inclusion
Pilot Test	30 individuals in China	China	Individuals experience in using software testing tools
Full Questionnaire Test	300 individuals in China	Jinan, Shandong, China	Individuals working as software testers in the information technology departments of companies from various industries

An initial investigation was carried out using a pilot study and purposive sampling method. In the initial phase, a pilot study that involved thirty participants from the software testing sector in China was conducted for one week. Pilot studies intended for preliminary survey or scale development typically require a minimum recommendation of 30 representative participants from the population of interest based on findings from Johanson’s study [29]. The survey study has determined a sample size of 300 individuals considering the time constraint. Most statisticians generally agree that a minimum sample size of 100 is necessary to obtain statistically meaningful results [30]. A sample size of 300 respondents has been selected to enable a broader exploration within the study in consideration of the large population of China. Another study conducted by Pallant stated that a sample size greater than 98 (N > 50 + 8(6)) is adequate, with 6 representing the number of independent variables [31].

Data for the study was collected through an online survey hosted on the WenJuanXing platform. It is important to emphasize that no private or sensitive data was collected. All survey data would be securely stored on WenJuanXing Forms and later downloaded in Excel format for further analysis. The data collection duration was one month. A comprehensive overview of the statistical analysis methods employed in this quantitative thesis is displayed in Table 6.

Table 6 Overview of techniques applied in quantitative analysis

Techniques	Purpose	Application
Descriptive Analysis (Bar Chart)	Summarize and present data distribution and relationships.	Display the distribution of categorical variables
Cronbach Reliability Test	Assess the internal consistency and reliability of a scale or instrument.	Evaluate the reliability of survey instruments or scales used in the study.
Pearson Correlation Test	Measure the strength and direction of a linear relationship between two continuous variables.	Examine the association between two quantitative variables in the study for the six hypotheses.
Multiple Regression Test	Analyze the relationship between a dependent variable and multiple independent variables.	Investigate the impact of multiple predictors on a specific outcome variable in the research.

FINDINGS AND RESULTS

Pilot test before survey distribution

The pilot test comprises gathering responses from 34 highly experienced professionals in China’s software testing sector through a questionnaire survey consisting of 27 questions. It was being conducted to validate the reliability of factors. Experienced individuals from the software testing field were selectively included to ensure the reliability of the questionnaire survey before opening the full test to all participants.

Table 7 outlines the examination of several factors. Each factor was presented by its respective set of items that are aimed at assessing the behavioral intention to utilize software testing tools. The reliability level of these factors was assessed using Cronbach’s alpha which is a scale to measure internal consistency.

Table 7 Reliability test run result in the pilot test

Variables	Number of items	Cronbach’s Alpha	Reliability Level
Functional Stability	3	0.856	Good
Reliability	4	0.743	Acceptable
Usability	5	0.816	Good
Security	5	0.868	Good

Maintainability	3	0.677	Questionable
Portability	4	0.849	Good
Behavioral Intention	3	0.877	Good

As shown in Table 7, all factors display good or acceptable reliability levels indicated by their respective Cronbach's alpha values except for the Maintainability factor. The Maintainability factor showed Cronbach's alpha of 0.677, falling below the commonly accepted threshold for reliability. This suggested a questionable level of internal consistency. A decision has been made not to include the Maintainability factor in the survey after careful consideration to uphold the overall reliability and validity of the assessment.

Descriptive analysis of survey responses

Figure 6 presents a bar chart depicting the distribution of 306 survey participants across various industries. It offers a comprehensive overview of the demographic representation within the surveyed population. It aids in understanding the utilization of software testing tools within each department, especially in the IT department.

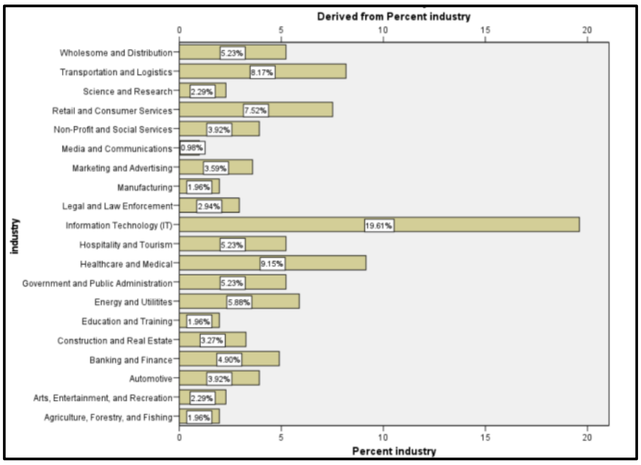


Figure 6 Distribution of survey participants across different industries

Quantitative analysis of survey responses

The results of the hypothesis testing as presented in Table 8 reveal positive and significant relationships between the listed variables and users' behavioral intention to adopt software testing tools. This analysis was conducted using the Pearson correlation test to examine hypotheses regarding one-to-one relationships.

Functional Stability (correlation: 0.369), Reliability (correlation: 0.511), Usability (correlation: 0.326), Security (correlation: 0.384), and Portability (correlation: 0.463) all demonstrate statistical significance, with p-values consistently below 0.05. This compelling statistical evidence suggests rejecting the null hypothesis. Reliability emerges as the most influential factor as it exhibits the highest correlation value in the Pearson correlation test. Even though Usability has the lowest correlation value, it still retains statistical significance with the test.

Table 8 Results of Pearson correlation test analysis

Hypothesis results	Significance value (p)	Pearson Correlation Value
H1- Functional Stability is positive and significantly related to the behavioral intention to use software testing tools: accepted	< 0.001	0.369
H2- Reliability is positive and significantly related to the behavioral intention to use software testing tools: accepted	< 0.001	0.511
H3- Usability is positive and significantly related to the behavioral intention to use software testing tools: accepted	< 0.001	0.326
H4- Security is positive and significantly related to the behavioral intention to use software testing tools: accepted	< 0.001	0.384
H6- Portability is positive and significantly related to the behavioral intention to use software testing tools: accepted	< 0.001	0.463

*Hypothesis 5 was not tested as the variable was removed from full test process.

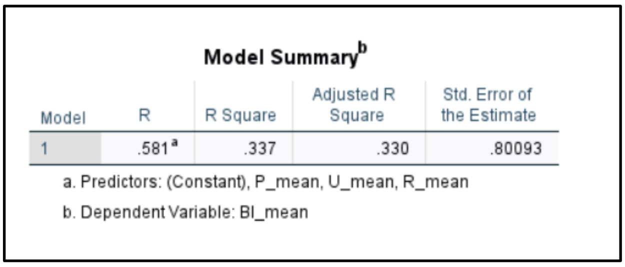
Table 9 displays the outcomes of multiple regression tests evaluating the influence of different independent variables on the intention to use software testing tools. The primary objective of multiple regression testing is to identify factors that can be used to derive the design framework. Functional Stability and Security exhibit coefficient beta values of 0.052 and 0.114 with corresponding p-values (Sig) of 0.396 and 0.057. Both p-values exceed the conventional significance threshold of 0.05. Therefore, both independent variables are considered insignificant contributors to the dependent variable. Subsequent regression analysis is performed with the exclusion of Functional Stability and Security due to their insignificance.

Table 9 First run multiple regression test results

Variables	Coefficients Beta	P-values	Evaluation
Functional Stability	0.052	0.396	Insignificant
Reliability	0.265	<0.001	Significant
Usability	0.157	0.011	Significant
Security	0.114	0.057	Insignificant
Portability	0.189	<0.001	Significant

The R-squared value from the second run of the multiple regression test after removing both non-significant factors Functional Stability and Security yields 0.337 as shown in Figure 7. The model summary result suggests that the remaining independent variables comprising Reliability, Usability, and Portability collectively explain 33.7% of the variance in the dependent variable. This indicates that the model can reasonably explain a moderate amount of the variation in the dependent variable. Although the remaining independent factors do not contain all the variability in the dependent variable, they still make a notable contribution to understanding and predicting the observed outcomes.

The ANOVA test reveals a significance value (Sig) less than 0.001 during the second run of the multiple regression test depicted in Figure 8. This suggests that the group of independent variables, including Reliability, Usability, and Portability, significantly contributes to explaining the variation in the dependent variable. The remarkably low p-value (less than 0.001) indicates a high level of confidence in rejecting the null hypothesis, thereby supporting the claim that at least one of the predictors in the model has a substantial impact on the dependent variable.



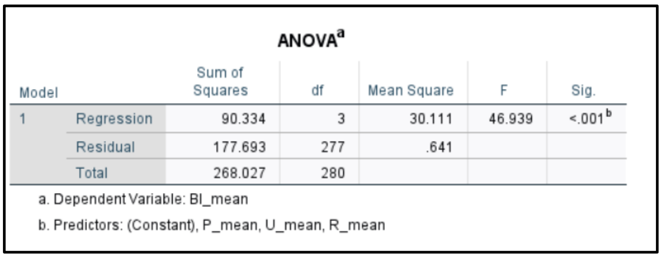
Model Summary[b]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.581[a]	.337	.330	.80093

a. Predictors: (Constant), P_mean, U_mean, R_mean

b. Dependent Variable: BI_mean

Figure 7 Results of model summary from the second run



ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	90.334	3	30.111	46.939	<.001[b]
	Residual	177.693	277	.641		
	Total	268.027	280			

a. Dependent Variable: BI_mean

b. Predictors: (Constant), P_mean, U_mean, R_mean

Figure 8 ANOVA results from the second run of multiple regression test

Table 10 displays the results of the second regression test, indicating that Reliability, Usability, and Portability have a significant influence on individuals' intention to use a software testing tool. Reliability stands out as a crucial factor. It shows a strong coefficient beta value of 0.312 and a very low p-value of <0.00 which indicates its importance in driving user intention. Both Usability and Portability also have moderate coefficient beta values of 0.177 and 0.223 and highly significant p-value of <0.001. These findings emphasize the significance of reliability, usability, and portability in shaping users' intention to use a software testing tool. The findings imply that leveraging these factors can help in formulating a software testing framework that is both effective and widely accepted in practical scenarios.

Table 10 Second run multiple regression test results

Variables	Coefficients Beta	P-values	Evaluation
Reliability	0.312	<0.001	Significant
Usability	0.177	0.004	Significant
Portability	0.223	<0.001	Significant

Conclusion

The research has effectively accomplished its primary objective of exploring the software quality factors that affect the intention to use software testing tools among the software testers in Jinan, Shandong, China. Both the Pearson correlation test and multiple regression test have identified the factors from the ISO/IEC 25010 model such as reliability, usability, and portability influenced the intention to use software testing tools among the software testers, thus aiding in the derivation of a software testing adoption framework.

There are two limitations of this research. Firstly, the reliance on a sample size of 300 responses raises concerns about the generalizability of the findings to the broader population. Additionally, it is important to acknowledge that not all respondents may possess a deep understanding of software testing tools despite their involvement in related industries, due to their working experience and exposure level to various software testing tools.

Future research endeavors should aim to broaden the context of the survey study by targeting a more diverse pool of participants across different regions of China. Furthermore, it is significant to target participants with solid experience in using software testing tools to ensure a detailed exploration of industry perspectives. Developers should incorporate integrated robust measures to ensure the reliability of testing tools [32], enhance user-friendly interfaces for improved usability [33], and enable seamless adaptability across diverse environments for optimal portability [34] when building software testing tools.

References

[1] A. Bertolino, “Software Testing Research: Achievements, Challenges, Dreams”, in Future of Software Engineering (FOSE '07), Minneapolis, 2007. https://doi.org/10.1109/FOSE.2007.25

[2] I. Hooda and R. S. Chhillar, “Software Test Process, Testing Types and Techniques”, International Journal of Computer Applications, vol. 111, no. 13, pp. 10-14, 2015. https://doi.org/10.5120/19597-1433

[3] H. Leung and L. White, “A Study of Integration Testing and Software Regression at the Integration Level”, in Proceedings. Conference on Software Maintenance 1990, San Diego, 1990. https://doi.org/10.1109/ICSM.1990.131377

[4] H. V. Gamido and M. V. Gamido, “Comparative Review of the Features of Automated Software Testing Tools”, International Journal of Electrical and Computer Engineering (IJECE), vol. 9, no. 5, pp. 4473-4478, 2019. https://DOI: 10.11591/ijece.v9i5.pp4473-4478

[5] A. Orso and G. Rothermel, “Software Testing: A Research Travelogue (2000–2014)”, in FOSE 2014: Future of Software Engineering Proceedings, 2014. https://doi.org/10.1145/2593882.2593885

[6] S. U. Farooq, “Gap between Academia and Industry: A Case of Empirical Evaluation of Three Software Testing Methods”, International Journal of System Assurance Engineering and Management, vol. 10, pp. 1491-1495, 2019. https://doi.org/10.1007/s13198-019-00899-2

[7] M. Catelani, L. Ciani, V. L. Scarano and A. Bacioccola, “Software Automated Testing: A Solution to Maximize the Test Plan Coverage and to Increase Software Reliability and Quality in Use”, Computer Standards & Interfaces, vol. 33, no. 2, pp. 152-158, 2011. https://doi.org/10.1016/j.csi.2010.06.006

[8] M. Böhme and S. Paul, “On the Efficiency of Automated Testing”, in FSE 2014: Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering, Hong Kong, 2014. https://doi.org/10.1145/2635868.2635923

[9] F. Okezie, Odun-Ayo and S. Bogle, “A Critical Analysis of Software Testing Tools”, Journal of Physics: Conference Series, vol. 1378, no. 4, pp. 1-11, 2019. https://doi.org/10.1088/1742-6596/1378/4/042030

[10] K. Sneha and G. M. Malle, “Research on Software Testing Techniques and Software Automation Testing Tools”, in 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), Chennai, 2017.

https://doi.org/10.1109/ICECDS.2017.8389562
[11] Amaralisa, "Non Functional Testing — A Detailed Overview", 2023a. [Website]. Available at: https://medium.com/@amaralisa321/non-functional-testing-a-detailed-overview-7e685bbeff61. Accessed on 10 February 2024.
[12] Amaralisa, "What are Non Functional Testing Types with Examples & Tools", 2023b. [Website]. Available at: https://medium.com/@amaralisa321/what-are-non-functional-testing-types-with-examples-tools-a775fbe91290#:~:text=Non%2Dfunctional%20testing%20is%20a,not%20available%20under%20functional%20testing. Accessed on 10 February 2024.
[13] N. Mulla and N. Jayakumar, "Role of Machine Learning & Artificial Intelligence Techniques in Software Testing", Turkish Journal of Computer and Mathematics Education, vol. 12, no. 6, pp. 2913-2921, 2021. https://doi.org/10.17762/turcomat.v12i6.5800
[14] K. Inçki, I. Ari and H. Sözer, "A Survey of Software Testing in the Cloud", in 2012 IEEE Sixth International Conference on Software Security and Reliability Companion, Gaithersburg, 2012. https://doi.org/10.1109/SERE-C.2012.32
[15] F. Khorram, J.-M. Mottu and G. Sunyé, "Challenges & Opportunities in Low-code Testing", in MODELS'20: Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, 2020. https://doi.org/10.1145/3417990.3420204
[16] F. H. Naeini and BalaKrishnam, "Usage Pattern, Perceived Usefulness and Ease of Use of Computer Games among Malaysian Elementary School Students", Research Journal of Applied Sciences, Engineering and Technology, vol. 4, no. 23, pp. 5287-5289, 2012. https://maxwellsci.com/print/rjaset/v4-5285-5297.pdf
[17] F. D. Davis, "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology", MIS Quarterly, vol. 13, no. 3, pp. 319-339, 1989. https://doi.org/10.2307/249008
[18] V. Venkatesh, M. G. Morris, G. B. Davis and F. D. Davis, "User Acceptance of Information Technology: Toward a Unified View", MIS Quarterly, vol. 27, no. 3, pp. 425-478, 2003. https://doi.org/10.2307/30036540
[19] A. A. Thunibat, A. Zahrawi, A. A. Tamimi and F. Al-tarawneh, "Measuring the Acceptance of Using Enterprise Resource Planning (ERP) System in Private Jordanian Universities Using TAM Model", International Journal of Information and Education Technology, vol. 9, no. 7, pp. 502-505, 2019. . https://doi: 10.18178/ijiet.2019.9.7.1254
[20] H. Rafique, A. O. Almagrabi, A. Shamim, F. Anwar and A. K. Bashir, "Investigating the Acceptance of Mobile Library Applications with an Extended Technology Acceptance Model (TAM)", Computers & Education, vol. 145, no. 103732, 2020. https://doi.org/10.1016/j.compedu.2019.103732
[21] A. K. Sharma and D. Kumar, "User Acceptance of Desktop Based Computer Software Using UTAUT Model and Addition of New Moderators", International Journal of Computer Science & Engineering Technology (IJCSET), vol. 3, no. 10, pp. 509-515, 2012. http://www.ijcset.com/docs/IJCSET12-03-10-043.pdf
[22] S. AlAwadhi and A. Morris, "The Use of the UTAUT Model in the Adoption of E-Government Services in Kuwait", in Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008), Waikoloa, 2008. https://doi.org/10.1109/HICSS.2008.452
[23] A. F. Aguirre, A. Villareal-Freire, R. Gil and C. A. Collazos, "Extending the Concept of User Satisfaction in E-Learning Systems from ISO/IEC 25010", in DUXU 2017: Design, User Experience, and Usability: Understanding Users and Contexts, Las Vegas, 2017. https://doi.org/10.1007/978-3-319-58640-3_13
[24] I. E. I. Fadhel, S. Z. B. S. Idrus, M. S. Y. Abdullah, A. A. E. A. Ibrahim, M. Omar and A. Khred, "A New Perspective of Web-Based Systems Quality Engineering Measure by Using

Software Engineering Theory (ISO 25010): An Initial Study", Journal of Physics: Conference Series, vol. 1529, pp. 1-14, 2020. https://doi.org/10.1088/1742-6596/1529/2/022004

[25] J. Estdale and E. Georgiadu, "Applying the ISO/IEC 25010 Quality Models to Software Product", in Systems, Software and Services Process Improvement, Bilbao, 2018. https://doi.org/10.1007/978-3-319-97925-0_42

[26] C.-h. Song and Y.-w. Sohn, "The Influence of Dependability in Cloud Computing Adoption", The Journal of Supercomputing, vol. 78, pp. 12159-12201, 2022. https://doi.org/10.1007/s11227-022-04346-1

[27] D. Sidik and F. Syafar, "Exploring the Factors Influencing Student's Intention to Use Mobile Learning in Indonesia Higher Education", Education and Information Technologies, vol. 25, pp. 4781-4796, 2020. https://doi.org/10.1007/s10639-019-10018-0

[28] P. C. Lai, "Design and Security Impact on Consumers' Intention to Use Single Platform E-payment", Interdisciplinary Information Sciences, vol. 22, no. 1, pp. 111-122, 2016. https://doi.org/10.4036/iis.2016.R.05

[29] G. A. Johanson and G. P. Brooks, "Initial Scale Development: Sample Size for Pilot Studies", Educational and Psychological Measurement, vol. 70, no. 3, pp. 394-400, 2010. https://doi.org/10.1177/0013164409355692

[30] P. B. Bullen, "How to Choose a Sample Size (For the Statistically Challenged)", 2013. [Website]. Available at: https://tools4dev.org/resources/how-to-choose-a-sample-size/. Accessed on 10 February 2024.

[31] J. Pallant, SPSS Survival Manual, London, 2020.

[32] Developer Experience Hub, "Software Testing and Quality Assurance: Best Practices and Strategies", 25 July 2023. [Website]. Available at: https://devxhub.medium.com/software-testing-and-quality-assurance-best-practices-and-strategies-7e2c8dc28262. Accessed on 10 February 2024.

[33] A. Darejah and D. Singh, "A Review on User Interface Design Principles to Increase Software Usability for Users with Less Computer Literacy", Journal of Computer Science, vol. 9, no. 11, pp. 1443-1450, 2013. https://doi.org/10.3844/jcssp.2013.1443.1450

[34] D. Foo, "Design Your Software for Portability", 19 November 2023. [Website]. Available at: https://danielfoo.medium.com/design-your-software-for-portability-00abcfbf5970. Accessed on 10 February 2024.