Role of Software Usability Tests and their Usefulness in Various Software Products

[1]Vinod Kumar, [2] Rajesh Kumar Maurya, [3]Tarun Kr. Sharma, [4]Devendra Kumar

[1, 2, 3]Assistant Professor, ABES Engineering College, Ghaziabad, Uttar Pradesh, India
[4]Professor, ABES Engineering Collage Ghaziabad, Uttar Pradesh, India

ABSTRACT
Ease of use shows a major role in the marketing of software products. Software usability tests are a planned approach to creating software systems that meets the needs of users in different environments with different computing experiences. The main goal of usability testing is to understand people who use software systems to know what people want. In educational institutions, moderators and learners must use usability test activities to help their educational organization. Some used it to provide important information to support procurement decisions; others used usability test data to understand where training and other learning efforts in the classroom should focus

KEYWORDS:— Testing, Software Testing Usability Testing, Human computer interaction, Criteria of Software, Advantage of Usability Testing, Disadvantage of Usability, Quality Consideration of Software.

1 .KEY CONCEPTS

Basic purpose of usability testing is on which way we have to use product in effectively. That determines the usability of software product. Here software testing has the capability to determine the evaluation of an attributes of a program or system and can determine the desired results or not[1]. That ensures the quality of the product through the verification and validation or reliability assessment. Use of software testing is to controlled the operating a system and verify the result. Normal and Non-normal state should be controlled state [2].

2. OVERVIEW OF USABILITY TESTING

This testing is the way where user can determine design of an application that meets the requirement of its users and allows us to productive work [3]. These test help to find the areas where person face problem of a product and recommendation for its refinements. The basic aim is to good understanding that a user touched with the artifact and to get better the result. Main objective of this test is to seem to be intended [4][5]. In this test, real users try to bring about characteristic goals and tasks, with software product under illegal conditions. The Researchers, stake holders, and development team take benefits from it[6][7].

3. QUALITY OF SOFTWARE

Software quality may be defined as practices for getting the desirable features of software product. Basically we have two main concepts for software product quality. It should be useful and practical. It should sustain the ability to apply repeatability, reliability, predictability. It has to be designed for phased section operation [8][9]. It must have the capability of being enhanced over time. It must be scalable across wide blend of industries and project types Defect management and quality attribute.

4. BENEFITS

The data sample size helps the uses to improve the usability. The rational data samples obtained after cross-validation improvise the usability of software. Sample size play very important role. Its results make for strong influence with design part and can have an important impact on product field. Such as in the difficult intertwine of system and with ordinary way of integrate for profit, software products into next developed systems, implication of software problems cannot always be probable, even in apparently ordinary term [21][22]

5. USER-BASED EVALUATION OF USABILITY TESTING

In this type of evaluations users directly involved. Users are free to do the task independently with a product, and free to explore it. If they found difficulties and errors, on the bases of this observation again design flow required to review and to improve it [13][14][15].

6. REFINE OF SOFTWARE QUALITY

- Requirement of focus orientation.
- Try to Capture Defects Consistently.
- The Modern-view shows the quality of a conformance of requirement.
- The product-view shows the quality of joined to innate features of the software product.
- The value-based view shows the quality of reliant on the quantity a customer is keen for pay.
- Higher quality software and shorter test cycles are possible when quality is the entire team's responsibility [16][17][18].

Table 1: Internal Quality Parameter

Test coverage		√	Yes
Testability		√	Hard
Portability		√	Somewhat
Thread-safeness		√	Hard
Conciseness		√	Somewhat
Maintainability		√	Hard
Documentation		√	Subjective
Power consumption		√	Difficult

7. DIMENSIONS OF QUALITY SOFTWARE

Software quality dimensions include accuracy, capability, communication, completeness, conformance, features, flexibility, serviceability, simplicity, stability, and structuredness. Consumer and producer attitudes about software quality only slightly differ. Exterior and Interior quality parameter discussed

.

Table 2: Dimensions of quality software

S.No.	Exterior Quality	Interior Quality	Future quality
1	Correctness	Efficiency	Flexibility
2	Reliability	Testability	Reusability
3	Usability	Documentation	Maintainability
4	Integrity	Structure	

8. CRITERIA FOR QUALITY SOFTWARE

The required level of service is specified according to a quality model. The software product quality model provided in ISO/IEC 9126-1 and also prepared ISO/IEC 25010 defines six quality characteristics: functionality, reliability, usability, maintainability, portability, and efficiency.

Table 3: Exterior Quality Parameter

Quality Parameter	User	Developer	Measurable
Space	1	1	1
Network usage	1	1	1
Stability	1	1	1
Features	1	1	1
Robustness	1	1	3
Ease-of-use	1	0	
Determinism	1	1	1
Security	1	0	2

N ----> represents 0, Y ----> represents 1, moderate--->2, rather--->3

9. FEATURE OF USABILITY

It is a Non-Functional testing technique. It keeps product or website in front of actual people from your object market. Intellectual and Physical expertise required For be taught the system. Time vital becoming moderately proficient to the System. The net amplify in production by use of the new system. Subjective assessment (usually in the form of survey on the new system)[10].

- Usability Test Process

10. USABILITY – UTILITY AND ITS BENEFITS

Basically both define usefulness of software usability shows how users of software product are able to learn and how use the capability of systems where as

utility refers to that a system that having the capability to accomplish a given task .Whereas Aspects of usability to learn ability, efficient to use handling the error and acceptability .which dealings with the genuine uses of product

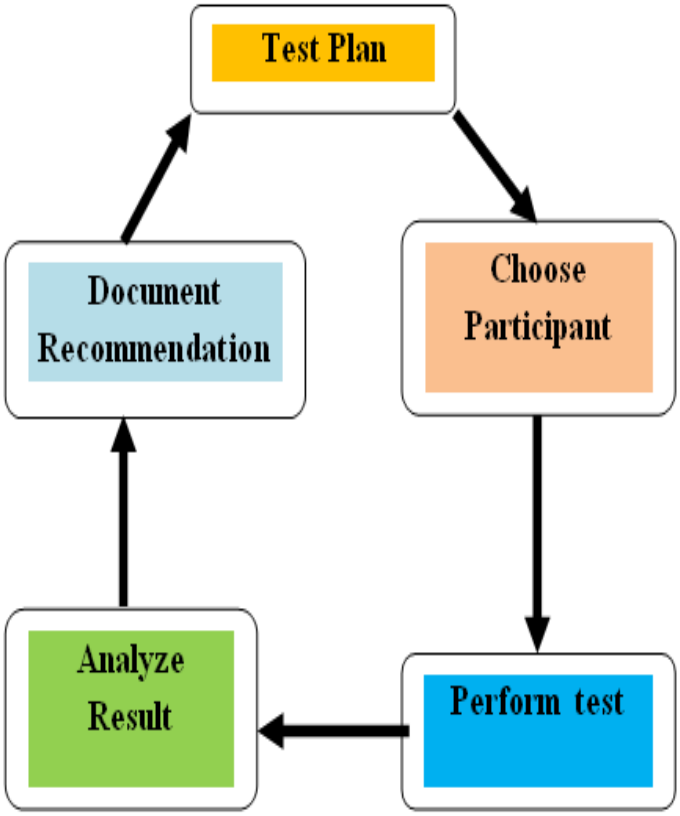

Figure 1: Usability Test Life Cycle

The goal researcher is going to complete. Concept of cooperation divides further into usability or utility. Figure1: represents usability test life cycle. Basically these terms are not exchangeable. Utility refers to the ability of the product to carry out tasks and more tasks the product is premeditated to carry out, the more service it has.[9][10]

Regard as a typical application of MS office is power point such programs gives a large variety of influential presentation and editing features, and many users have to learn and remember many hidden keystrokes for perform well. These types of applications have high and necessary become too familiar with the "quirks" of the software - and not report a possible error or usability issue. These types of problem can be alleviated by including user testers in the test team [23][24].They functionality but low usability means users required to expend a much time and effort for learn and use. Such as a simple application like a calculator some time very easy to use but may not offer more utility. Here Usability testing determine how it can easy is for users to perform a tasks [9]. And it save the time for the company as well as for users and gives a better experience to users.

12. BENEFITS

'Usability Testers' to the test team can have a very positive effect on the team itself. Several times I have seen that testers can also help to refocus the testers and increase their awareness to usability issues, by providing a fresh viewpoint. Provide and share their expert knowledge - training the testers to the background and purpose

of the system [11][12].Provide a "realistic" element to the testing, so that test scenarios are not just "possible permutations [25].

13. CONCLUSION

This article addresses it should include prior in SDLC for reducing resistance to changes in a toughened user-interface. There is a need for self-regulating usability-evaluation software-products to avoid the enticement to ignore problems and discharge the product. Many categories to dependent relative way should be service in usability- testing because subjective dimension is not always consonant with user recital; and Even though usability testing at the later stage of growth may not blow software changes, it is useful to point out areas where training is needed to conquer deficiency in the software product. Software product design issues should be with the user's viewpoint.

REFERENCES

[1] Jedlitschka, Andreas, et al. "Empirical practice in software engineering." *Perspectives on the future of Software Engineering*. Springer, Berlin, Heidelberg, 2013. 217-233.

[2] Vetro, A., Ognawala, S., Fernandez, D. M., & Wagner, S. (2015, May). Fast Feedback Cycles in Empirical Software Engineering Research. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering (ICSE)* (Vol. 2, pp. 583-586).

[3] Münch, J., & Schmid, K. (2013). *Perspectives on the Future of Software Engineering*. Springer-Verlag Berlin Heidelberg.

[4] Sharafi, Z., Sharif, B., Guéhéneuc, Y. G., Begel, A., Bednarik, R., & Crosby, M. (2020). A practical guide on conducting eye tracking studies in software engineering. *Empirical Software Engineering*, 1-47.

[5] Shull, F., Singer, J., & Sjøberg, D. I. (Eds.). (2007). *Guide to advanced empirical software engineering*. Springer Science & Business Media.

[6] Chulani, S., Boehm, B., & Steece, B. (1999). Bayesian analysis of empirical software engineering cost models. *IEEE Transactions on Software Engineering*, *25*(4), 573-583.

[7] Obilade, T. T. (2020). Implications of Similarities in Instructional Design, Learner Interface Design and User Interface Design in Designing a User-Friendly Online Module. In *Multicultural Instructional Design: Concepts, Methodologies, Tools, and Applications* (pp. 240-259). IGI Global.

[8] Ndamase, Z., & Padayachee, I. An Investigative Research On Users' Perceptions of Information Systems Service Quality at The University of KwaZulu-Natal. In *Research Conference of Richfield Graduate Institute of Technology* (p. 102).

[9] Wilson, J. P., & Campbell, L. (2020). ISO 9001: 2015: the evolution and convergence of quality management and knowledge management for competitive advantage. *Total Quality Management & Business Excellence*, *31*(7-8), 761-776.

[10] Gonzalez-Feliu, J., Chong, M., Vargas-Florez, J., Osorio-Ramirez, C., Piatyszek, E., & Quilche Altamirano, R. (2020). The Maturity of Humanitarian Logistics against Recurrent Crises. *Social Sciences*, *9*(6), 90.

[11] Anand, A., Kaur, J., & Inoue, S. (2020). Reliability modeling of multi-version software system incorporating the impact of infected patching. *International Journal of Quality & Reliability Management*.

[12] Wibowo, A., & Davis, J. (2020, February). Requirements Traceability Ontology to Support Requirements Management. In *Proceedings of the Australasian Computer Science Week Multiconference* (pp. 1-9).

[13] Petrov, A., Popova, E., & Petrov, A. (2020). Some Aspects of a Software

Reliability Problem. *arXiv preprint arXiv:2003.02590*.
[14] Shaw, G., Whelan, M. E., Armitage, L. C., Roberts, N., & Farmer, A. J. (2020). Are COPD self-management mobile applications effective? A systematic review and meta-analysis. *NPJ primary care respiratory medicine*, *30*(1), 1-10.
[15] Nygård, M. (2020). Evaluating the Usability of the Finnish Translation of NHL 20.
[16] Lindblom, J., Alenljung, B., & Billing, E. (2020). Evaluating the User Experience of Human–Robot Interaction. In *Human-Robot Interaction* (pp. 231-256). Springer, Cham.
[17] Blomqvist, S., & Detterfelt, B. (2020). Real Time Integrated Tools for Video Game Development: a usability study.
[18] Wichansky, A. M. (2000). Usability testing in 2000 and beyond. *Ergonomics*, *43*(7), 998-1006.
[19] Au, F. T., Baker, S., Warren, I., & Dobbie, G. (2008, January). Automated usability testing framework.In *Proceedings of the ninth conference on Australasian user interface-Volume 76* (pp. 55-64).
[20] Weiss, S. (2003). *Handheld usability*. John Wiley & Sons.
[21] Naqvi, B., Seffah, A., & Abran, A. (2020). Framework for examination of software quality characteristics in conflict: A security and usability exemplar. *Cogent Engineering*, *7*(1), 1788308.
[22] Mohammed, M. A., Talib, A. M., & Al-Baltah, I. A. (2020). Metrics and Models for Evaluating the Quality of ERP Software: Systematic Mapping Review. In *Metrics and Models for Evaluating the Quality and Effectiveness of ERP Software* (pp. 1-27). IGI Global.
[23] Goyal, S. (2020). Comparison of Machine Learning Techniques for Software Quality Prediction. *International Journal of Knowledge and Systems Science (IJKSS)*, *11*(2), 20-40.
[24] Boucher, A., & Badri, M. (2018). Software metrics thresholds calculation techniques to predict fault-proneness: An empirical comparison. *Information and Software Technology*, *96*, 38-67.
[25] Kumar, L., Krishna, A., & Rath, S. K. (2017). The impact of feature selection on maintainability prediction of service-oriented applications. *Service Oriented Computing and Applications*, *11*(2), 137-161.