https://doi.org/10.48047/AFJBS.6.14.2024.6613-6617

Optimizing Software Quality: Learnability Metrics and Approaches

B. Thillaieaswaran[1], Dr.C.Nageswari, [2] K.Rajkannan [3]

[1]Assistant Professor, [2]Associate Professor, [3]Assistant Professor
[1,3]School of Computer Science and engineering,
[2]School of Allied Health Sciences
Galgotias University, Greater Noida, Uttar Pradesh, India
b.thillaieaswaran@galgotiasuniversity.edu.in , nageswari@galgotiasuniversity.edu.in,
k.rajkannan@galgotiasuniversity.edu.in

Article History
Volume 6, Issue 14, Aug 2024
Received:03 July 2024
Accepted : 31 July 2024
Published :22 Aug 2024
doi:10.48047/AFJBS.6.14.2024.6613-6617

Abstract: Software quality is often defined as a measurable attribute of a software system, closely linked to usability. Usability, in turn, refers to how easily and effectively a software product can be used by humans. This encompasses ease of use, which involves subjective assessments of user experience, and effectiveness, which relates to the user's ability to achieve desired outcomes. The software development lifecycle aims to identify and optimize the activities involved in creating software. The success of many software systems is heavily dependent on their usability. While some development models neglect usability considerations, poor usability often leads to software failure. In this study, we explore various methods for enhancing software quality through the application of usability metrics, highlighting their critical role in improving overall software performance.

Keywords: Software Excellence, System Quality, Software Performance, Product Quality, Quality parameters, Quality models

I. INTRODUCTION

Usability extends beyond the mere appearance of a user interface; it encompasses how effectively users interact with the software. Key usability attributes include learnability, efficiency, user retention over time, error rate, and overall satisfaction. Additionally, user experience quality (UX) and delight (DOD) can be evaluated through various methods. In most applications, developers collect user-related data to gain insights into how the software is used, which helps in refining and enhancing the user experience. This feedback is crucial for continuous improvement and delivering superior service. However, it is imperative that developers respect user privacy and avoid misuse of this information. There are also numerous innovative applications for information mining that leverage this data to further enhance software quality and user satisfaction.

II. REVIEW OF LITERATURE

Lot of workable models available promising software quality assurance. But some are giving priority to Usability metrics and some are not. Here we are going to discuss only popular models. Neilsen's usability model [19] deals with the usability characteristics such as Learnability, Efficiency, Satisfaction Memorability and Error. ISO 9241-11[20] usability model deals with Effectiveness, Efficiency and satisfaction. MGQM [18] usability deals with Simplicity, Accuracy, Time taken, Feature, Safety and Attractiveness, PACMAD [21] Extend PACMAD [22] usability models deal with effectiveness, efficiency, Satisfaction, Learnability, Memorability, Errors and Cognitive load. QUIM [23] usability model deals with Effectiveness, Efficiency, Productivity Satisfaction, Learnability, Safety, Truthfulness, Accessibility, Universalizability and usefulness. MUSIC [24] usability model deals with Effectiveness and Efficiency. SUMI [25] usability model deals with Questionnaire method for quality assessment of measuring users' perception of the usability software. In addition to the usability parameters some models use usability factors like utility, adaptability, playfulness, simplicity etc. So, the usability can be attributed by effectiveness and easiness. The main challenge in our work is to identify which of the above models uses interaction as an important parameter to the development.

2.1 Common interaction

Common interaction types in software include instructing, conversing, manipulating, and exploring. Instructing involves issuing commands and selecting options, while conversing simulates a dialogue with the system. Manipulating refers to interacting with objects by adjusting or controlling them, and exploring involves navigating through a virtual or physical space. The choice of interaction type depends on device capabilities and software design, aiming to maximize user satisfaction. These interaction methods contribute to the concept of User Experience (UX), which can be assessed through the Quality of Experience and involves UX design principles, as well as the Degree of Delight (DOD). Although this discussion does not delve deeply into measuring these aspects, it is important to recognize the significant role that interaction design plays in shaping effective software experiences.

2.2 Interaction styles

In Human-Computer Interaction (HCI), the focus is on how users communicate and interact with computer systems. This interaction can be broadly categorized into four main types: Command Language Interaction, Form Filling Interaction, Menu Selection, and Direct Manipulation. Conventional interaction methods, such as command language and form filling, typically offer detailed insights into user behaviour and preferences. In contrast, methods like menu selection may provide more limited information. Some applications leverage a combination of these interaction methods to

gather comprehensive and valuable data about users, enhancing the overall user experience.

2.3 Existing implementation

2.3.1 Interviews

In this method, a set of sample questions is created to cover all relevant attributes of the application, with the user providing answers to evaluate the system's usability. For instance, in the research study titled "Usability Study of a Web-Based Platform for Home Motor Rehabilitation," conducted by Jorge Luis Perez Medina, Mario Gonzalez, Henry Mauricio Pilco, Karina Beatriz Jimenez Vargas, Patricia Acosta Vargas, Sandra Sanchez Gordon, Tania Calle Jimenez, Danila Esparza, and Yves Rybarczyk, the authors categorized usability into several dimensions, including SYSUSE, INFOQUAL, INTERQUAL, and OVERALL. This study, which received support from IBM, provides valuable insights for developers and contributes significantly to the field of health systems. The authors extend their gratitude for the support and recognition of their work.

2.3 Characteristics table

A table will be created to document a small-scale experiment involving novice, intermediate, and expert users on a widely-used platform. The table will record values before and after user interactions. The results of the experiment are expected to be insightful, revealing the significant impact of interaction at each stage of the user experience.

2.4 Usability specification table

A table is prepared and given in a research work “Usability basics for software developers” by Xavier ferre and Natalia jurist, Helmut windl, larry Constantine explained. To check usability by interaction and sample attributes may be minimum acceptable level, planned target level, best possible level and Observed results.

2.4 Technologies Employed in Common Applications

In the realm of software development, various technologies are employed to enhance and support common applications. These technologies are integral to achieving functional efficiency, user satisfaction, and overall system performance. Below are some key categories and examples of technologies frequently used:

2.4.1. Front-End Technologies:

HTML/CSS: Essential for structuring and styling web content, HTML (HyperText Markup Language) and CSS (Cascading Style Sheets) form the backbone of web development.

JavaScript: A scripting language used to create interactive and dynamic content on web pages. Libraries and frameworks like React, Angular, and Vue.js enhance its capabilities.

Responsive Design Frameworks: Tools like Bootstrap and Foundation ensure applications are accessible and functional across various devices and screen sizes.

2.4.2. Back-End Technologies:

Programming Languages: Languages such as Java, Python, Ruby, and PHP are used to develop server-side logic and handle data processing.

Frameworks: Frameworks like Django (Python), Spring Boot (Java), and Ruby on Rails (Ruby) streamline back-end development by providing pre-built modules and conventions.

Databases: Relational databases like MySQL, PostgreSQL, and SQLite, as well as NoSQL databases like MongoDB and Cassandra, manage and store data efficiently.

2.4.3. Middle ware Technologies:

APIs (Application Programming Interfaces): APIs enable different software systems to communicate with each other, facilitating integration and data exchange.

Message Brokers: Tools like RabbitMQ and Apache Kafka handle asynchronous communication between different parts of an application or between different services.

2.3.4. Cloud Technologies:

Cloud Platforms: Services such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform offer scalable resources for hosting, storage, and computing.

Serverless Architectures: Platforms like AWS Lambda and Google Cloud Functions enable developers to build and run applications without managing server infrastructure.

Development and Deployment Tools:

Version Control Systems: Tools like Git and platforms like GitHub and GitLab help manage code changes and collaboration among developers.

Continuous Integration/Continuous Deployment (CI/CD): Technologies like Jenkins, Travis CI, and CircleCI automate testing and deployment processes, ensuring consistent and reliable releases.

Security Technologies:

Encryption: Protocols such as SSL/TLS and encryption libraries protect data in transit and at rest.

Authentication and Authorization: Technologies like OAuth, JWT (JSON Web Tokens), and identity management systems secure user access and permissions.

User Experience (UX) and Usability Tools:

UX Design Tools: Applications like Sketch, Figma, and Adobe XD facilitate the design and prototyping of user interfaces.

Usability Testing Tools: Platforms such as User Testing and Hotjar provide insights into user behaviour and interaction, helping to refine and improve application usability.

By leveraging these technologies, developers can create robust, user-friendly applications that meet diverse needs and perform efficiently in various environments. The choice of technology depends on the specific requirements of the application, including its functionality, scalability, and user experience goals.

III. PROPOSED SYSTEM

Fig.3.1 shows our proposed model to developed a software application with the help of usability at all levels.

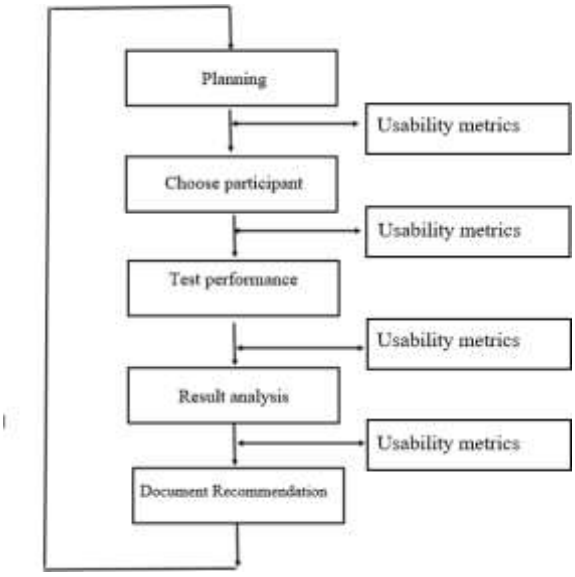

Fig 3.1 Proposed model

IV. Conclusion and Future Work

To support our work, we have conducted a manual experiment with three

kinds of students (novice, intermediate and expert level of) higher secondary school students for a popular online platform. To make the experiment for analysis purpose we have prepared sample questions and given values from 0 to 9 for each attribute and showed as sample table 3.1. In future the same work can be done and atomized for all other usability parameters instead of doing manual experiment

Table 3.1: ICT on Learnability

Learnability Effectiveness		
Novice	Intermediate	Expert
0	5	9
1	4	8
0	4	9
3	3	8

Learnability is nothing but how easy to learn the main system functionality and proficiency to complete the task.

V. REFERENCES

1 Laura Carvajal, Ana M. Moreno, Maria Isabel Sánchez-Segura, and Ahmed Seffah, Usability through Software Design, IEEE Transactions On Software Engineering, Vol. 39, No. 11, November2013.

2 Md AlamgirKabir and Muaan Ur Rehman and Shariful Islam Majumdar, An Analytical and Comparative Study of Software Usability Quality Factors - Usability Model in Software Engineering Literature , IEEE, 978-1-4673-9904-3/16/,2016.

3 Carl J. Mueller, Dan Tamir, Oleg V. Komogortsev, Liam Feldman, An EconomicalApproach to Usability Testing, 33rd Annual IEEE International Computer Software and Applications Conference, 2009.

4 Deepak Gupta and Anil Ahlawat, Taxonomy of GUM and Usability Prediction Using GUM Multistage Fuzzy Expert System, The International Arab Journal of InformationTechnology, Vol. 16, No. 3, May2019.

5 Ohamma d Reza HeidariIman and Abbas Rasoolzadegan, Quantitative Evaluation of Software Usability with a Fuzzy Expert System, 5th International Conference on Computer and Knowledge Engineering (ICCKE), 2015.

6 LAI Xiangwei, ZHOU Yanhuiand ZHANG Weiqun, Software Usability Improvement: Modeling, Training and Relativity Analysis, IEEE, Second International Symposium on Information Science and Engineering,2009.

7 Shashank Sharma and Sumit Srivastava, Generalized Software Quality Assurance Technique for Maintenance parameter Evaluation, IEEE,2014.

8 Luis F. Mendivelso, Kelly Garcés and Rubby Casallas, Metric centre and technology independent architectural views for software comprehension, Journal of software engineering and research development, Vol 12,No.2,Feb 2018.