

Application-based Usability Evaluation Metrics

Hanaa Bayomi¹, Noura A.Sayed², Hesham Hassan³, Khaled Wassif⁴

Faculty of Computers and Artificial Intelligence, Cairo University, Cairo, Egypt^{1, 3, 4}
Faculty of Computers and Artificial Intelligence, Akhbar El Youm Academy, Cairo, Egypt²

Abstract—Testing is one of the vital stages in the software development life cycle (SDLC). Usability testing is a very important field that helps the applications be easily used by the end-users. Because of the importance of usability testing, a metrics has been developed to help in measuring the usability through converting the main qualitative usability attributes in ISO to quantitative steps that provide the developer a framework to follow in developing to achieve usability of their applications and helps the tester with a checklist and a tool to measure the usability percentage of their application. The framework provides a set of steps to achieve the usability attributes and answers the question of how you could measure this attribute with the defined steps. The framework results in a 95% average accuracy in the high-rate application and a 59% average accuracy in the low-rate application. Finally, the framework is programmed in a tool to measure the usability percentage of the application through a checklist and provides a scheme to help the developer achieve the best results in usability.

Keywords—Usability; human-computer interaction; evaluation; quantitative attributes; testing

I. INTRODUCTION

Testing is one of the important stages of the SDLC. Usability testing is one of the non-functional testing types [1]. Usability is one of the quality dimension that evaluates the quality and usefulness of applications [2]. Usability is a quality attribute that determines how easily and simply user interfaces can be used. So, usability is the interface's necessary condition for application's survival [3]. If an interface is difficult to use and the users get lost, they will leave. No user would be bothered with reading a manual first before using an interface or a website. Therefore, usability is a very vital element of an application interface.

Accordingly, a framework has been developed to convert the qualitative usability attributes into qualitative steps to help the developer to have simple steps to achieve the highest usability and helps the tester to have a checklist to easily evaluate the application usability. According to Nielson, usability can be defined as a method for improving the design process [4]. Usability is assessed based on six dimensions, which include learnability, memorability, efficiency, effectiveness, error rate, and user satisfaction. [5]. Usability testing uses the black box testing technique [1]. Our framework can be performed by the tester at the system and acceptance levels. As well, usability is one of the nonfunctional requirements of any software, which is one of the core areas of research in the field of human-computer interaction [6].

Now-a-days, both Web and mobile applications are considered the most two popular types of applications.

Usability has been defined as a key component in the overall quality of a software product, and research shows that usability can determine the success or failure of a software system [7]. Usable software systems are not only more efficient, accurate, and safe, but also much more successful and several studies have shown the benefits of incorporating usability evaluation in the process of software development. Therefore, usability evaluation has become an important research field [7].

The remainder of this paper is organized as follows: Section II is related work, which provides background information about usability and testing techniques; Section III is methods, which describe how the research is designed and carried out; Section IV is usability attributes, which describe the attributes in well-defined steps; Section V is testing references and questions, which describe the usability attributes in testing questions; and Section VI is results, which describe the findings.

II. RELATED WORK

Nielsen (1994) defined usability as a quality attribute that measures how user interaction can be used as a method for improving ease-of-use, efficiency, and satisfaction [4]. Standard ISO 9241-11 defines usability as "the extent to which a product can be used by specific users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use" [7]. It was further elaborated that there are other important concepts related to usability, such as user, goal, effectiveness, efficiency, satisfaction, and context of use [8].

Usability of software applications is one of the core areas of research in the field of human-computer interaction. There were many methods to inspect, test, and inquire about an application's usability [7], [9].

Making the application usable doesn't mean that it will be easy. Some applications must be a little bit difficult to use to help in reaching the target of the application's usage for example e-learning applications [10]. The e-learning applications' difficulty must be specified according to the learning level of the student to meet the target of increasing the student's learning curve [11]. One of the very effective ways of evaluating the e-learning applications' usability is the heuristic evaluation method [12], [13]. Heuristic evaluation can also be used to measure other types of applications because it has been shown to be effective in inspecting errors [14], [15], [16].

Before evaluating any application, the latest issues must be reviewed and the most commonly occurring errors arise

related to your application type [17]. To change the qualitative attributes to quantitative, the quantitative attributes must be revised [18]. Mobile and web applications are the most commonly used types of applications today. Both of those types can be accessed via mobile, which is the most convenient device to hold most of the time, so we must concentrate on mobile application usability [19], [20], [21], [22], [23].

Most of the papers focus on updating the usability attributes by adding some new attributes or deleting some of existing attributes or changing the way of the attribute evaluation [24]. Some of the papers are updating the usability main categories introduced in the ISO/IEC 25022 [3], [6]. Some papers are creating new modules to be remotely evaluating the applications through user usage [25]. Some papers focus on one of the usability evaluation techniques [26] [27]. Some papers focus on analyzing the Human computer interaction or usability attributes according to a specific application type [28], [29]. There is some approaches and frameworks that improve issues tracking [30], [31], [32].

According to the ISO, those are the documented standards presented in Table I [6].

TABLE I. ISO USABILITY ATTRIBUTES REFERENCES

ISO reference number	Usability attribute category
ISO 9241-11	Usability Definitions and concepts, replacing the 1988 version of ISO 9241-11
ISO 9241-220	Processes for enabling, executing and assessing human-centered design within organizations, replacing the earlier ISO TR 18529
ISO/IEC 25066	Common industry Format for Usability — Evaluation Reports
ISO/IEC 25022	Measurement of quality in use, (includes measures of effectiveness, efficiency and satisfaction), replacing ISO TR 9126-4
ISO/IEC 25023	Measurement of system and software product quality, (includes measures for usability attributes), replacing ISO/IEC TR 9126-2 and ISO/IEC TR 9126-3

This research highlighted the following usability gaps: first, there is a problem in evaluating the usability of the application's interface. Second, developers do not focus on the simplicity of the interface as they focus on the functionality of the application because the functionality is easy to develop as there are no clear guidelines for evaluating the interface usability. Third, there is no balance between user experience and interface simplicity. For example, learning applications mustn't be so easy or so difficult to meet the goal of increasing the learning curve. Finally, there is no clear qualitative borderline to be applied by the developer and tested by the tester. To fill those gaps, a framework has been developed to describe the usability attributes in fine steps and helps in achieving the best usability of the application.

III. METHODOLOGY

To convert usability attributes in the ISO from qualitative to quantitative steps, this is done by searching for how the developer could achieve each attribute in well-defined steps

and asking usability experts what the points they take into consideration to make the application usable enough for the user. The 20 developers and testers were asked about the frequently repeated problems that they face while developing and testing the applications. After collecting the required data, usability attributes have been chosen from the ISO Usability standards. Some of those attributes were merged and some have been added to create an effective usability framework. In the framework, the qualitative chosen attributes have been converted to detailed steps to help the developer and the tester in their practical work. Then, one or two questions is assigned for every attribute to be asked by the tester for evaluation and tested on the application to measure the usability.

The results have been developed by choosing two applications, one with a high rate (amazon) and the other with a low rate (waradly) to test the effectiveness of the framework developed. The first is Amazon, which has a global web application and also a mobile application with a rate of 4.3 out of 5, and the other is Waradly, which is an Egyptian e-commerce site with an intermediate rate, but they don't have a mobile application. To test our framework, three main tasks have been chosen that are performed by any e-commerce application, which are searching, seeking product data, and purchasing.

Finally, a tool has been developed using our framework to measure the usability percentage of the application through a checklist and provide a scheme to help the developer to achieve the best results in usability (<http://abuem.com/>).

IV. USABILITY ATTRIBUTES

Accordingly, steps have been assigned for each attribute in each category to be measured quantitatively, merging, and adding some attributes to them.

A. Effectiveness

1) *Tasks completed and objectives achieved:* The applications mainly provide the user with services or products. To complete the task, you have to use a scale from 1 to 10 according to the feasibility analysis conducted by the business analyst. To achieve effectiveness in tasks completed and objectives achieved, set reasonable objectives according to each business environment, choose the best developing environment to achieve these objectives (desktop, mobile, or web application), set tasks, specify for each task input and output, specify for each task severity and priority, and order the tasks according to the scale set by feasibility analysis. Here is an example to calculate the rest %. Subtract the finished tasks from the number of all tasks to give you the rest of the tasks. Then divide the result by the number of all tasks and multiply by 100 to give you the whole percent of the rest of the tasks.

2) *Errors in a task:* To achieve effectiveness in errors in a task attribute, the errors that may arise at each task individually must be predicted; choose an efficient programming language to minimize the predicted errors, fix the unpredicted errors, and document them with their solution

to be fixed automatically if they arise again, according to the literature.

3) *Tasks with errors*: To achieve the effectiveness in tasks with errors attribute, make sure that the errors in each stage will be solved individually, don't move to the next task unless the previous task is clear of any errors, and after testing, the total number of errors mustn't exceed 25% of all the tasks to deliver the system to the end-user.

4) *Task error intensity*: To achieve the effectiveness in task error intensity attribute, task errors must be classified according to task complexity, error intensity must be set from 1 to 5 based on the severity and priority set by a tester, and the factors that cause the error must be identified in order to solve it or give the user a hint on how to deal with it in order to reduce the intensity of its existence.

B. Efficiency

1) *Task time*: To achieve the Efficiency in Tasks time attribute, the developer must set the shortest way for the user to make the task to achieve the goal in the easiest way, set the minimum developing time for a task according to its complexity without the existence of any errors, set the maximum developing time for a task according to its complexity with the existence of all errors that could happen, and complete each task within the maximum and minimum of it.

2) *Time efficiency*: To achieve the Efficiency in Tasks efficiency attribute, the developer must choose a task at a time to set its input and desired output. Set the best programming language to achieve your target and choose the best algorithm according to its big O to achieve the best time efficiency.

3) *Cost-effectiveness*: To achieve the efficiency in cost-effectiveness attribute, a feasibility study must be developed based on the hardware and software needs (developing and testing environment) and choose the best platform to satisfy user needs, including user supplies to support most of the devices he can afford in software and hardware.

4) *Productive time ratio*: To achieve efficiency in the productive time ratio attribute, the developer must develop a timing strategy for ending the project and create criteria to set time for each feature based on the employee performance to achieve the optimal standard above and more quantity.

5) *Unnecessary actions*: To achieve the efficiency in unnecessary actions attribute, after finalising your code, all of the code must be revised to make sure that you choose the best development path and choose the quickest way to make the function with the minimum number of steps.

6) *Fatigue*: To achieve the Efficiency in Fatigue attribute, the developer must be aware of the problems that occur while developing that affect the deadlines and add them to the delivery date, and be prepared for the cases that make the system load slowly or go down.

C. Satisfaction

1) *Overall satisfaction*: To achieve satisfaction in the overall satisfaction attribute, the project must be delivered

before the deadline, have well-defined requirements that have one meaning, and involve small prototypes to involve the users while developing.

2) *Satisfaction with features*: To achieve the satisfaction with features attribute, continuous messages must be displayed to inform the user of the current processing, block all the expected errors, and minimize the needed resources for working.

3) *Discretionary usage*: To achieve the Satisfaction in Discretionary usage attribute, the developer must think about all the cases where the user needs to use the application, and according to each case, the developer must satisfy the user's needs, satisfy the client's needs, and satisfy the market's needs. Finally, the developer must develop all the ways to achieve each target, for example, adding a speech-to-text feature and chatting to enter data.

4) *Feature utilization*: To achieve the satisfaction in feature utilization attribute, each feature must be explained to the user, choose the simplest way to develop each feature; they must choose the shortest number of steps to end a task; and the page must contain at most one objective or task.

5) *Proportion of users complaining*: To achieve satisfaction in the overall satisfaction attribute, the application must be too clear to reduce user inquires, provide a chat pot as a supporter in any problem, and provide quick support in a large problem.

6) *Proportion of user complaints about a particular feature*: To achieve the satisfaction in the overall satisfaction attribute, the application must help the user at the beginning by giving him a demo for each feature, giving the feature a definition and descriptive logo, and making the feature complete on one page with fine input and several steps. Then put the main features on the home page and their shortcuts on other pages. Finally, try to use AI in the application to tell the user if he is making the wrong decision.

7) *User trust*: To achieve the satisfaction in the overall satisfaction attribute, the application must: get permission from the user before saving or using his data, give a full explanation for why their data is needed as audio or text, solve all the user's problems most simply; then support the user with all the knowledge he may need; and involve the user as much as possible while developing to see the prototypes.

8) *User pleasure*: To achieve the satisfaction in user pleasure attribute, the application must develop every single task easily with fewer errors, give users continuous explanations and motivation messages, choose cheerful designs and colors to make the user happy, and ask the user for their feedback continuously.

9) *Physical comfort*: To achieve the satisfaction in physical comfort attribute, a backup must be taken for all your work to make the user feel safe, use UPS to assure that the data will be safe, and use the simplest way to develop the application.

D. Appropriateness Recognizability

1) *Readability*: To achieve the appropriate level of recognizability in the readability attribute, the icons must be readable to allow the other attributes like learnability, accessibility, and so on to be achieved. So, the font of the icon must be as big as you can to minimize the number of objects that exist on one page, whether it is a mobile or a web application.

2) *Description completeness*: To achieve appropriateness recognizability in the description completeness attribute, user stories must be wisely described to satisfy the developer and have one meaning based on the user's point of view, market needs, and recommendation of the best way to develop (interface architecture).

3) *Demonstration coverage*: To achieve the appropriateness recognisability in demonstration coverage attribute, the developer should develop many application types and many targets of applications at the same field to be aware of all the vulnerabilities of developing, and while developing, the developer must study all the pros and cons of any way related to developing.

4) *Entry point self-descriptiveness*: To achieve the appropriateness recognizability in the entry point self-descriptiveness attribute, requirements must be determined based on user stories that describe each scenario that could happen by the user to make the most effective decision while developing, and each feature must be described in the simplest way to make it easy to develop by a beginner developer.

E. Learnability

1) *User guidance completeness*: To achieve the learnability in user guidance completeness attribute, the project mustn't be launched until the user manual is available in the form of steps.

2) *Entry fields defaults*: To achieve the learnability in entry fields defaults attribute, the right option must be very clear to the user to avoid errors in data entry fields.

3) *Error message*: To achieve the learnability in an error message attribute, every problem in every step the user can do, an error message must be designed to clarify the error and help him avoid it.

4) *Understandability*: To achieve the learnability in the understandability attribute, every step must be clear and mustn't have a double meaning as possible.

5) *Self-explanatory user interface*: To achieve the Learnability in Self-explanatory User Interface Attribute, interface parts must be mapped to the objects used in our lives to be easy to understand, recognize, and recall.

F. Operability

1) *Operational consistency*: To achieve operability in the operational consistency attribute, the page mustn't contain many tasks or any other information that doesn't relate to the main task (each page for one purpose).

2) *Message clarity*: To achieve the operability in the message clarity attribute, any message that appears to the user must be very clear, and if the developer has time, he must add an option to let the user see the last message again when it disappears.

3) *Functional customizability*: To achieve the operability in the functional customizability attribute, the application must have fewer mandatory fields in each task as much as possible for less time consumption and make a detailed manual consisting of steps to make the application very easy.

4) *User interface customizability*: To achieve the operability in the user interface customizability attribute, flexibility must be given to the user to change the places of the objects on the interface as well as the colors to make the application more comfortable.

5) *Monitoring capability*: To achieve the operability in the monitoring capability attribute, the application must provide the user with an indication when he gets closer to the end of his task and provide the user with full statistics of the application usage and resource consumption. Then, inform the user about memory and resource usage, help the user to reduce memory consumption, and don't ask the user for too much data to be saved or user in one task.

6) *Undo capability*: To achieve the operability in the undo capability attribute, the user must have the ability to go back with clear action and help them to recognize it in many situations.

7) *Understandable categorization of information*: To achieve operability in understandable categorization of information attributes, the application must provide the user with a thorough explanation of the input data and why it is required, and then inform the user if his data will be used.

8) *Appearance consistency*: To achieve operability in the appearance consistency attribute, the system must have one design on every page, stick to simple and realistic symbols and designs to relate them to our lives, avoid attaching unrelated things to the application, and avoid putting too many details on one page even if they are related to the same task.

9) *Input device support*: To achieve the operability in the input device support attribute, the application must work on cheap resources, don't make the application need many hardware devices to work properly, and try the application before delivery to the customer on different category devices according to the budget.

G. User Error Protection

1) *Avoidance of user operation error*: To achieve user error protection in the avoidance of user operation error attributes, the application mustn't allow the user to input wrong data and must give a full explanation for the user's errors.

2) *User entry error correction*: To achieve user error protection in the User entry error correction attribute, the application has to give the user a full description of the error

type and reason for its happening and give the user suggestions to correct his error.

3) *User error recoverability*: To achieve user error protection in the user error recoverability attribute, the application must allow the user to go back one step to make the user feel safe, give the user a description for each step he makes to increase awareness, and provide the user with a short manual that explains all the steps of the application.

H. User Interface Aesthetics

To achieve user interface aesthetics in the appearance aesthetics of user interfaces attribute, the developer has to choose the best comfortable color for the eye, the uploaded pictures must be of high resolution and minimum size to decrease the time of loading and increase the font of the data as much as possible to satisfy the user.

I. Accessibility

1) *Accessibility for users with a disability*: To achieve user interface aesthetics in the appearance aesthetics of user interfaces, the developer must be aware of all the types of users that will use the application and their disabilities, like color blindness, wheelchairs, etc.

2) *Supported languages adequacy*: To achieve user interface aesthetics in supported languages adequacy attribute, the application must support the global language and the regional language, and in the future, you can add the most important languages ordered by priority.

V. TESTING REFERENCES AND QUESTIONS

Some will need to be revised from the requirements documentation, while others will need to be tested on the application. Here is a table that identifies which attributes need to be tested from the application only and which attributes need to be revised from what has been approved in the requirements documentation and sets some suggested questions for each attribute described in Table II.

TABLE II. DESCRIPTIVE QUESTION OF USABILITY ATTRIBUTES

Attributes	Requirements documentation or Application usage	Questions to ask for an evaluation
Tasks completed and Objectives achieved	Requirements documentation	Are all the system tasks according to the system requirements is done? Do you achieve all the objectives of the system according to the required documentation?
Errors in a task	Application usage	Do you have any errors in any tasks? Then try it
Tasks with errors	Application usage	After testing each task, is all the tasks that have errors exceed 25% of all the tasks?
Task error intensity	Application usage	Is the task error frequently happening?
Task time	Application usage	Is each task takes the

		minimum time it must take?
Time efficiency	Application usage	Do you develop each task to be done at the minimum processing time?
Cost-effectiveness	Requirements documentation	Are you stick to the feasibility analysis you made before starting?
Productive time ratio	Requirements documentation	Is every second in the application is useful to the user?
Attributes	Requirements documentation or Application usage	Questions to ask for an evaluation
Unnecessary actions	Application usage	Is every step in each task a must and has a specific target?
Fatigue	Application usage	Taking into consideration the maximum number of users at a time works with stability?
Overall satisfaction	Application usage	Dose all the usability requirements tested to achieve high satisfaction?
Satisfaction with features	Application usage	Does the simplicity of each feature exist?
Discretionary usage	Application usage	Can you use the system anytime anywhere?
Feature utilization	Application usage	Do all the features the system support are mandatory?
The proportion of users complaining	Application usage	Will the user need to call customer support or see the manual many times while usage?
The proportion of user complaints about a particular feature	Application usage	Does every feature need to be seen first from a manual?
User trust	Application usage	Does the system makes the user feel that he/ she is secure enough to trust the system?
User pleasure	Application usage	Does the system make the user pleased by the system?
Physical comfort	Application usage	Does the system allow comfortable positions while usage?
Readability	Application usage	Do all the pages is seen clearly by higher ages?
Description completeness	Application usage	Do you make a user manual to explain the system in steps or develop a demo to help the user to see the steps of any task?
Demonstration coverage	Application usage	Does the system has an explanation coverage?
Entry point self-descriptiveness	Application usage	Is every step described in its meaning, target, and how to the user without using the manual?

User guidance completeness	Application usage	Do you create a manual to help the user in explaining the system?
Entry fields defaults	Application usage	Do you explain the default inputs to the user?
Error message	Application usage	Do you give messages every time the user could make an error?
understandability	Application usage	Is the system easy to use and helps user memory to recover quickly?
Self-explanatory user interface	Application usage	Is the interface explain itself by mapping the tasks to the user knowledge?
Operational consistency	Application usage	Does the page contain many tasks?
Message clarity	Application usage	Does the information in all the messages that appears to the user clearly described?
Functional customizability	Application usage	Do all the fields in every step are mandatory?
User interface customizability	Application usage	Does the user have the flexibility to change the interface appearance?
Attributes	Requirements documentation or Application usage	Questions to ask for an evaluation
Monitoring capability	Application usage	Do the system support monitoring for the user errors to help him in the long run?
Undo capability	Application usage	Do you allow the user to go back whenever he wants?
Understandable categorization of information	Application usage	Do you inform the user of the structure of the data and why the application uses it?
Appearance consistency	Application usage	Do the system parts have consistency?
Input device support	Application usage	Does all the input devices available to the user while working with the system?
Avoidance of user operation error	Application usage	Do you take into consideration all the errors that can be done by the user and block them?
User entry error correction	Application usage	Do you design error messages to help the user with their errors?
User error recoverability	Application usage	Do you allow the user to solve the errors he/she made?
Appearance aesthetics of user interfaces	Application usage	Does the interface explain every step?
Accessibility for users with a disability	Application usage	Is the system helps users with different disabilities to use it easily?
Supported languages adequacy	Application usage	Do the system support many languages mainly the global

		language and the user's first language?
--	--	---

VI. RESULTS AND DISCUSSION

A. Results

Three main tasks have been chosen according to their importance to any e-commerce users which are search, seek product data and purchase to test them on two applications one with high rate (amazon) and the other with low rate (waradly). Assume that the attributes that need to be revised from the requirement documentation are all true. The results have been calculated by testing framework attributes on the two application simultaneously through asking the previous questions of each attribute. The answers will be true or false. If answer is true, so the attribute is applied on the application. If answer is false, so the attribute is not applied on the application. Every true will be converted to 1 grade and every false will be converted to 0 grade. Finally, all the grades of all the attributes are added, then divided by the total number of attributes which are 44, and multiplied by 100 to get the usability percentage of each task usability. For example, after testing the searching task on amazon application by opening the application and answer all the usability attributes on the searching task to check the existence of every attribute, the result was 42 attributes were true from total 44 attributes. The result was calculated by calculating total true attributes, divided by total number and multiply 100 ($42/44*100$) to give the result 95%. All the final results are presented in Table III after testing all the usability attribute on every e-commerce task then compare the usability results of every application to proof that the framework can differentiate between the usability of the high rate and the low rate applications.

TABLE III. USABILITY TESTING RESULTS

E-commerce Task	Results after checking the attributed existence on every e-commerce task	
	High rate Application(<i>Amazon</i>)	Low rate Application(<i>Waradly</i>)
Searching	95 %	49 %
Seeking product data	100 %	61 %
Purchasing	95 %	59%

B. Discussion

Most of the previous work papers focus on updating the needed attributes that is used in testing the applications and test them to proof their idea or taking an existence framework and upgrade the way of evaluation to get better results. But in this research, the framework focus on describing all the usability attributes from qualitative title into quantitative detailed steps that helps the developer and tester on their daily work.

VII. LIMITATIONS OF THE STUDY

For future improvement, this research contains some limitations as setting usability attributes for each application type according to the application categories. Also, differentiating between the general attributes that could be applied on all applications and the specific application types.

Finally, setting weights for each attribute to clarify the importance of each attribute and order them according to the attribute weights.

VIII. CONCLUSION

After testing the framework on two e-commerce applications, one with a high rate and the other with a low rate on three main tasks, which are searching, seeking product data, and purchasing, our framework proves that a high rate (Amazon) resulted in 95% in the searching task while a low rate (Waradly) resulted in 49%. Also, it proves that a high rate (Amazon) resulted in 100% in the seeking product data task while a low rate (Waradly) resulted in 61%. Finally, a high rate (Amazon) resulted in 95% in the purchasing task while a low rate (Waradly) resulted in 59%.

So that means our framework proved that Amazon has a high rate and Waradly has a low rate, which means it can measure the usability percentage and categorises the application usability. The testers can use the framework in testing their applications of any type as the attributes are generic, which means they can measure any type of application.

ACKNOWLEDGMENT

First and foremost, I feel always indebted to Allah the Most Beneficent and Merciful. Words can never suffice to reflect my sincere appreciation and gratitude to Prof. Dr. Hesham Hassan, Prof. Dr. Khaled Wassif, and Dr. Hanaa Bayomi.

REFERENCES

- [1] M. A. Umar, "Comprehensive study of software testing: Categories, levels, techniques, and types," *International Journal of Advance Research, Ideas and Innovations in Technology*, 2019.
- [2] Ashraf Mousa Saleh and Roesnita Binti Ismail, "Usability Evaluation Frameworks Of Mobile Application: A Mini-Systematic Literature Review," in *The 3rd Global Summit on Education GSE 2015At: Kuala Lumpur, MALAYSIA*, 2015.
- [3] ASHRAF SALEH, ROESNITA BINTI ISAMIL, NORASIKIN BINTI FABIL, "EXTENSION OF PACMAD MODEL FOR USABILITY EVALUATION METRICS USING GOAL QUESTION METRICS (GQM) APPROACH," *Journal of Theoretical and Applied Information Technology*, 10th September 2015.
- [4] Nielsen, *Usability Inspection Methods*, New York: NY: John Wiley and Sons, 1994.
- [5] Shabana Shareef and M.N.A. Khan, "Evaluation of Usability Dimensions Of Smartphone Applications," *International Journal of Advanced Computer Science and Applications*, Vols. Vol. 10, No. 9, 2019.
- [6] Nigel Bevan, Jim Carter, Jonathan Earchy, Thomas Geis, Susan Harker, "New ISO Standards for Usability, Usability Reports and Usability Measures," M. Kurosu (Ed.): *HCI 2016, Part I, LNCS 9731*, p. 268–278, July 2016.
- [7] ANUBHA GULATI and SANJAY KUMAR DUBEY, *CRITICAL ANALYSIS ON USABILITY EVALUATION TECHNIQUES*, *International Journal of Engineering Science and Technology (IJEST)*, 2012.
- [8] Nigel Bevan, Jim Carter, Jonathan Earchy, Thomas Geis and Susan Harker, "ISO 9241-11 Revised: What Have We Learnt About Usability Since 1998?," in M. Kurosu (Ed.): *Human-Computer Interaction, Part I, HCII 2015, LNCS 9169*, Springer International Publishing Switzerland, 2015, p. 143–151.
- [9] Bevan N. & Macleod M., "Usability measurement in context.," in *Behaviour and Information Technology*, 1994, p. 132–145.
- [10] Bimal Aklesh Kumar & Munil Shiva Goundar & Sailesh Saras Chand, "Usability guideline for Mobile learning applications: an update," *Education and Information Technologies*, June 2019.
- [11] Sidra Shahid and Muhammad Shabbir Abbasi, "Usability Testing of an E- Learning System:A Comparative study of two Evaluation Techniques," *IOSR Journal of Computer Engineering*, vol. 16, 2014.
- [12] Bimal Aklesh Kumar & Munil Shiva Goundar & Sailesh Saras Chand, "Usability Guideline for Mobile Learning Applications," *Education and Information Technologies*, Jan 2019.
- [13] Estela Aparecida, Oliveira Vieira, Aleph Campos Da Silveira and Ronei Ximenes Martins, "Heuristic Evaluation on Usability of Educational Games: A Systematic Review", *Informatics in Education*, Vols. Vol. 18, No. 2, p. 427–442, 2019.
- [14] Bimal Aklesh Kumar and Priya Mohite, "Usability of mobile learning applications: a systematic literature review," *J. Comput. Educ.*, 2017.
- [15] Muhammad Salman Bashir and Amjad Farooq, "EUHSA:: Extending Usability Heuristics for Smartphone Application," *IEEE Access*, vol. 4, 2017.
- [16] RUYTHER PARENTE DA COSTA, EDNA DIAS CANEDO, RAFAEL TIMÓTEO DE SOUSA, JR2 (Member, IEEE), ROBSON DE OLIVEIRA ALBUQUERQUE and LUIS JAVIER GARCÍA VILLALBA3, "Set of Usability Heuristics for Quality Assessment of Mobile Applications on Smartphones," *IEEE Access*, 2017.
- [17] Carmelo Ardito1 & Maria De Marsico2 & Davide Gadia3 & Dario Maggiorini3 & Ilaria Mariani & Laura Ripamonti & Carmen Santoro, "Special Issue on Advances in Human-Computer Interaction," in *Multimedia Tools and Applications*, Springer Science+Business Media, LLC, part of Springer Nature 2019, 2019, p. 13353–13359.
- [18] Anju Gautam, Prashant Sahai Saxena & Savita Shiwani, "Quantitative analysis of usability issues of Ecommerce portal," *Journal of Statistics and Management Systems*, pp. 681-692, 2017.
- [19] P. WEICHBROTH, "Usability of Mobile Applications: A Systematic literature study," *IEEE Access*, 2020.
- [20] Jakub Myka, Agnieszka Indyka-Piasecka, Zbigniew Telec, Bogdan Trawiński, Hien Cao Dac, "Comparative Analysis of Usability of Data Entry Design Patterns for Mobile Applications," *research gate*, Jan 2019.
- [21] Azham Hussain, Nor Laily Hashim, Nazib Nordin and Hatim Mohamad Tahir, "A METRIC-BASED EVALUATION MODEL FOR APPLICATIONS ON MOBILE PHONES," *Journal of ICT*, vol. 12, p. 55–71, 2013.
- [22] Luis Cayola and José A. Macías, "Systematic guidance on usability methods in user-centered software development," *Information and Software Technology*, 2018.
- [23] Karima Moumane, Ali Idri and Alain Abran, "Usability evaluation of mobile applications using ISO 9241 and ISO 25062 standards," *springer plus*, p. 15, 2016.
- [24] Azham Hussain, Emmanuel O.C. Mkpjojiogu, Nur Hafiza Jamaludin and Somia T.L. Moh, "A usability evaluation of Lazada mobile application," *American Institute of Physics*, 2017.
- [25] Nouzha Harrati, Imed Bouchrika, Abdelkamel Tari and Ammar Ladjailia, "Automating the Evaluation of Usability Remotely for Web Applications via a Model-Based Approach," *First International Conference on New Technologies of Information and Communication (NTIC)*, 2015.
- [26] Ger Joyce, Mariana Lilley, Trevor Barker and Amanda Jefferies, "Heuristic Evaluation for Mobile Applications: Extending a Map of the literature," in *Advances in Usability, User Experience and Assistive Technology*, 2019, pp. (pp.15-26).
- [27] Christiane Gresse von Wangenheim, Talita A. Witt, Adriano Ferreti Borgatto, Juliane Vargas Nunes, Thaisa Cardoso Lacerda, Caroline Krone, Laís de Oliveira Souza., "A Usability Score for Mobile Phone application based on Heuristics," *International Journal of Mobile Human Computer Interaction*, 2016.
- [28] Zahra Al Mahdi, Vikas Rao Naidu, and Preethy Kurian, "Analyzing the Role of Human Computer Interaction Principles for E-Learning solution design," *Smart Technologies and Innovation for a Sustainable Future, Advances in Science, Technology & Innovation*, 2019.

- [29] Azham Hussain and Emmanuel O.C. Mkpojiogu, "Usability Evaluation Techniques in Mobile Commerce Applications: A Systematic Review," Proceedings of the International Conference on Applied Science and Technology, 2016.
- [30] Mamta Pandeya, Ratnesh Litoriyaa, and Prateek Pandeya, "Application of Fuzzy DEMATEL Approach in Analyzing Mobile App Issues," Programming and Computer Software., 2019.
- [31] Renuka Nagpal, Deepti Mehrotra, Pradeep Kr. Bhatia and Arun Sharma, "FAHP Approach to Rank Educational Websites on Usability," International Journal of Computing and Digital Systems, 2015.
- [32] Mamta Pandey, Ratnesh Litoriya, Prateek Pandey, "Identifying Causal Relationships in Mobile App Issues An Interval Type 2 Fuzzy DEMATEL Approach," wireless personal communication, 2019.