Towards Measuring User Experience based on Software Requirements

Issa Atoum[1], Jameel Almalki[2], Saeed Masoud Alshahrani[3], Waleed Al Shehri[4]
The World Islamic Sciences and Education University, Amman, Jordan[1]
Department of Computer Science, College of Computer in Al-Leith, Umm Al-Qura University, Makkah, Saudi Arabia[2, 4]
Department of Computer Science, College of Computing and Information Technology, Shaqra University, Shaqra, Saudi Arabia[3]

Abstract—**User Experience (UX) provides insights into the users' product perceptions while using or intending to use an application. Software products are known for complexity and changeability, starting from requirements engineering until the product operation. Users often evaluate software UX based on a prototype; however, UX is semantically embedded in the software requirements, a crucial indicator for project success. The problem of current UX evaluation methods is their dependence on the actual involvement of users or experts, a time-consuming process. First, this paper builds a benchmark dataset of UX based on textual software requirements crowdsourcing several UX experts. Second, the paper develops a machine learning model to measure UX based on the dataset. This research describes the dataset characteristics and reports its statistical internal consistency and reliability. Results indicate a high Cronbach Alpha and a low root mean square error of the dataset. We conclude that the new benchmark dataset could be used to estimate UX instantly without the need for subjective UX evaluation. The dataset will serve as a foundation of UX features for machine learning models.**

Keywords—User experience; benchmark dataset; requirements engineering; UX evaluation; software engineering

I. Introduction

The User Experience (UX) is a term used to indicate personnel perceptions and resulting emotions from systems or services [1]. Some of the concepts related to UX are included in the definition of Quality in Use (QinU); the user perceptions could result from a system or software's hedonic and pragmatic qualities. Very often, software UX is related to nonfunctional requirements and usability. Therefore, UX is a broader concept that includes usability, user satisfaction, emotions, and perceptions during the interaction [2]. On the other hand, the UI is considered a mechanism of functionality, usability, reliability, and satisfaction [3]. Consequently, usability is influenced by two orthogonal aspects, GUI and UX. However, the focus of this paper is evaluating the UX even if no UI is already built. Therefore, post evaluations of UX after product release (e.g., [4]) are not considered in this study.

The UX evaluation depends on components or factors of UX models. Some of the most commonly cited factors are proposed by Morville [5], known as the honeycomb model. The honeycomb model is based on balancing context, content, and users. Morville's honeycomb consists of seven factors: useful, usable, desirable, findable, accessible, credible, and valuable. However, Morville's model is generic to any product and is not focusing on software products. Recently, questionnaire-based approaches [6][7], heuristics models [8] [9], and hybrid models [13] were proposed for UX modeling and evaluation.

One of the frameworks that lay a foundation on software requirements and UX requirements is the UX-aware framework of Kashfi *et al.* [10]. The UX-aware framework shows that UX requirements are embedded quality requirements that describe the end user's satisfaction or pleasure of using the software. However, UX is generally evaluated based on a prototype(or a release); therefore, productive UX evaluation models could be used early during the requirements development [11]. Regrettably, the unavailability of measurement metrics or a benchmark dataset breaks the early evaluation. The automation of UX measurement is widely disregarded due to its complexity. In this era, an agent was proposed to automate UX testing for specific predefined tasks of objects finding scenarios in a game application [11].

Machine learning has been extensively adopted in many domains to predict and estimate target variables; however, a useful machine learning model depends on robust and reliable datasets. To the best of our knowledge, there are no existing datasets specialized for UX based on software requirements.

Therefore, the research gap is related to the unavailability of automated UX evaluation methods, which will result in that UX evaluators spending extensive efforts. Therefore, the UX evaluators are regretted using manual UX evaluation, including conducting surveys. Worst of all the evaluators must use an existing prototype to evaluate the UX (using existing evaluation methods). As a result, requirements engineers are not fully aware of any early UX software requirements before getting any prototype. Lacking large datasets of software requirements are still a major challenge for proper cost-effective and rapid application development[12]. Moreover, the elimination of UX-compliant requirements results in UX neglection in the final product [13] or poor resource planning [14]. Therefore, the automation of UX evaluation lacks datasets and proper evaluation models.

The main objective of this paper is to build a dataset that could be used for UX evaluation using machine learning techniques solely relying on textual requirements before an actual software gets developed. We employed four UX experts to annotate user requirements to a widely accepted scale[1], the User Experience Questionnaire (UEQ) [15][16][17]. The proposed benchmark dataset is a PROMISE-based dataset[18], one of the non-commercial requirements datasets. The research uses the latest expanded PROMISE dataset[19], part of the Open Science Tera-PROMISE repository upgraded earlier [18].

The contributions of this study are critical to software requirements engineering and user acceptance success. The prepared benchmark dataset eases the UX evaluation using machine learning models instead of traditional approaches such as questionnaires or heuristic models. This study is also important for software developers to track the software quality over the software life cycle. Therefore, the overall advantage of using an automated UX system is important for both software consumers and developers. Thereby, saving time and efforts of software developers and providing early feedback to software consumers. The main research question that addresses the UX measurement is as follows:

How UX evaluation could be automated at the early stages of requirements engineering using machine learning?

The paper structure is as follows. Section 2 discusses related works. Then, the methodology of building the benchmark dataset is illustrated in Section 3. Next, Section 4 describes the benchmark dataset reliability and provides preliminary results of machine learning methods. After that, we show the implications of the proposed dataset and its limitations in Section 5. Finally, we conclude the paper in Section 6.

II. Related Work

The UX has been evaluated in several models. Morville [5], one of the leaders in UX, proposed the honeycomb model, which consists of seven factors: useful, usable, desirable, findable, accessible, credible, and valuable. A system is considered usable if its needs are delivered in a simple way considering the user learning curve. If the system fulfills the user's needs it is considered useful. However, a system is considered desirable based on its design and attractiveness. If further information is needed about the system it should be findable, easy to navigate. The system should be accessible even to a user with disabilities. Therefore, an application is considered credible if it is trustworthy. Integrating all of these factors provides a valuable system. However, Morville's model is generic to any product and is not focusing on software products.

Generally, UX evaluation models could be classified into three categories: questionnaire-based approaches, heuristics models, and hybrid frameworks. Questionnaire-based models [6], [7], [15], [20] are known for their simplicity in evaluation and aggregation. In such methods, the users are responsible for the system evaluation with a few questions based on their perceived use of products. For example, the benchmark of the User Experience Questionnaire (UEQ) [15] uses several question items to evaluate the UX. Each item of the UEQ consists of a pair of terms with opposite meanings (e.g., 'not understandable' vs. 'understandable', 'efficient' vs. 'inefficient') on a 7-point Likert scale that each ranges from -3 to +3. With UEQ, several users would evaluate a system prototype to calculate key performance indicators(KPIs) for each software application under study (e.g., [6]). Such KPIs are based on UEQ scores and simple average and summation statistics. Recently a questionnaire survey was developed to measure usability, usefulness, and satisfaction of a chatbot UX [21]. However, the approach of 19] was customized for conversational agents.

On the other hand, heuristics models depend on rules or checklist items prepared and evaluated by experts in the specific service domain. In this category, Yeratziotis and Zaphiris [9] proposed a set of rules (heuristics) to support human-computer interaction experts to evaluate website accessibility. Similarly, online travel agency applications' UX has been evaluated formally with 8 stages [2]; however, this model[2] is application-dependent.

UX has been evaluated in the context of students' applications. The UX has been seen from the angle of usefulness and effectiveness of students teaching systems [22] [23]. For example, Krouska *et al.* [23] proposed a set of rules for students' misconception of HTML to understand the user (student) experience while using an e-learning system. Based on the student experience with the system the proposed model —based on the repair theory— was able to suggest a student learner path. Although heuristic models enable experts to key in needed knowledge to help in UX evaluation, they are generally applied to certain application domains where evaluation checklists exist.

Hybrid models combine the previous methods to measure the UX such as measuring the physiological aspects of users during software usage [24]. The multimodal deep learning model UX framework of Hussain *et al.* [24] depends on sentiment analysis, user feedback, and visual analysis of user action using sensors that detect user's objects on the screen. Koonsanit and Nishiuchi [25] proposed a framework to measure software UX based on facial expression recognition and machine learning; however, their focus is on product sentiment analysis rather than measurement of UX from software requirements [26]. A similar model was proposed by Li and Liu [27] to analyze user eye-movement tracking along with user testing methods to suggest actions for a better user experience. Furthermore, a UX framework for business intelligence (BI) systems interfaces was proposed by Eriksson and Ferwerda [28] to support users' desires and data evaluation. The evaluation of their framework utilizes several KPIs: utility, usability, visual attractiveness, and hedonic quality that covers the whole system development lifecycle. Jang and Han [29] proposed a framework for UX understanding in blockchain services. The UX is defined based on the literature in UX generality and blockchain technological aspects. More specific UX frameworks were proposed for educational games. Leong *et al.* [21] considered the game flow, player context, usability, and learnability along with psychometrically to build an educational UX measurement model.

[1]Google Scholar shows 1,301 citations for UEQ paper (2008): 11/2021.

The literature shows that the gap is the current methods are time-consuming and subjective. UX measurement depends on persons with different expertise level questionnaires and product trustworthiness which are widely subjective [30]. To our knowledge, this could be the first paper that heights the importance of requirements UX evaluation before an application gets developed based on machine learning models.

III. Methodology

The proposed methodology is based on textual software requirements that are considered the initial source of software UX [31], [32]. The source dataset for software requirements is the expanded PROMISE dataset [19], as described in TABLE I.

UEQ metrics are the foundation of the benchmark dataset because it is a widely used UX evaluation method for software products[6]. Specifically, we employed the short UEQ (UEQ-S) model that has eight items [6]. The core UEQ primary constructs are as illustrated in TABLE II [6].

In addition, these constructs are represented using eight class labels in a score ranging from 1 (minimum) to 7 (maximum) for each class label, as shown in TABLE III. For example, a requirement item with a score of 5 for 'label 1' is considered supportive, while an item of score 1 is considered obstructive. The neutral value is (3). Note that the same requirement item could have eight scores simultaneously.

The annotation scheme methodology is depicted in Fig. 1. We selected four experts of UX who have more than five years of experience in UX design. First, they were interviewed online to know the robustness of their work. Then, we explain to them the objectives of the work and the UEQ measurement scales (Step 1). After the explanation, we draw a random 100 requirements (approximately 10% of the dataset) from the PROMISE dataset with different applications (Step 2). The experts studied the complete set of requirements for each application. Next, they were allowed to discuss how to classify requirements to the eight UX scales. After that, they were left to do the annotation alone (Step 3). During the data reconciliation process (Step 4), we again choose another random 50 annotated requirements (not the previous ones), and experts were allowed to discuss discrepancies (if any). It was found that all 50 pairwise scores were acceptable having an absolute error value of 1 to 2. Finally, we got four Excel sheets for the exact requirements classified into eight labels, each with a scale in the range 1-7.

TABLE I. ATTRIBUTES OF EXPANDED PROMISE DATASET [19]*

Attribute	**Description**	**Total**
Project-ID	**The project IDs range from 1 to 49. Projects are generic domains such as shopping and universities.**	**49**
Requirement-Text	**Project textual requirements at the analysis stage of analysis and design.**	**969**
Class	**Functional or non-functional. Requirements where nonfunctional requirements are a set of 13 subcategories.**	**13 (F(1), NF(12))**

*Functional (F), and Nonfunctional (NF) requirements are part of this dataset.

TABLE II. CORE COMPONENTS OF THE UEQ SCALES (ALSO DESCRIBED IN AL-HUNAIYYAN *ET AL.* [37])

Construct	**Meaning**
Attractiveness	The product should be pleasurable, user-friendly, and enjoyable
Efficiency	Perform tasks with the product in a fast manner and pragmatically
Perspicuity	The product plain to the understanding primarily because of clarity, and easiness to learn
Dependability	The product services that can be trusted within time and meets users' expectations
Stimulation	Using the product encourages its use due to being exciting and motivating
Novelty	The product should be pioneering, inspired, and creatively designed

TABLE III. UEQ-S LABELS[6]

Label ID	**Quality Category**	**Negative Word**	**Positive Word**
Label 1	Pragmatic Quality	obstructive	supportive
Label 2	Pragmatic Quality	complicated	easy
Label 3	Pragmatic Quality	inefficient	efficient
Label 4	Pragmatic Quality	confusing	clear
Label 5	Hedonic Quality	boring	exciting
Label 6	Hedonic Quality	not interesting	interesting
Label 7	Hedonic Quality	conventional	inventive
Label 8	Hedonic Quality	usual	leading-edge

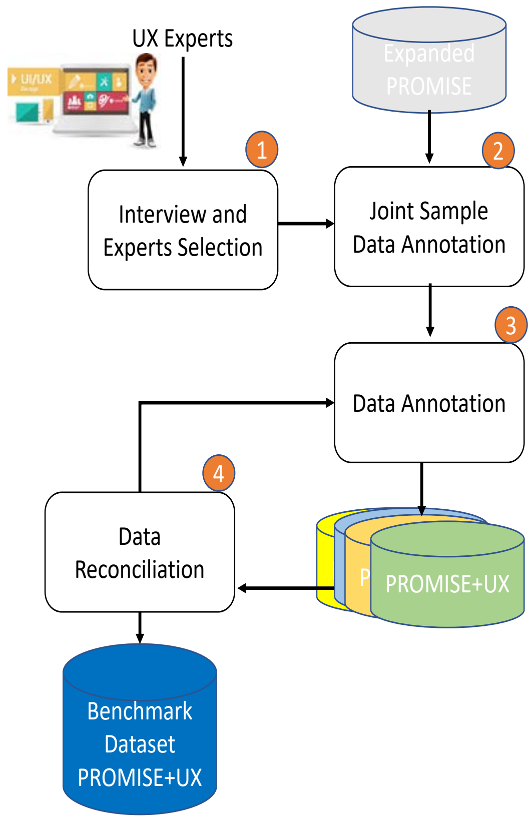
UX Experts
Expanded PROMISE
1
2
Interview and Experts Selection
Joint Sample Data Annotation
3
Data Annotation
4
Data Reconciliation
PROMISE+UX
Benchmark Dataset PROMISE+UX

Fig. 1. Proposed Annotation Scheme Methodology.

The reliability and consistency of the collected dataset were tested with Cronbach Alpha and the root means square error (RMSE) to measure the differences between experts' rating scores. Cronbach alpha is applicable since we have weighted items (1 to 7), which could be used to explain the proportion of variance between different experts [33]. On the other hand, the RMSE measures the average magnitude of errors in annotation scores between experts. It is a desirable measure as it gives relatively more weight to errors of large magnitude that need to be eliminated.

IV. Results and Discussions

We illustrate the reliability of the proposed benchmark dataset and preliminary UX evaluation results on this dataset.

A. *Benchmark Dataset Reliability*

The user experience team previously hired for this experiment is playing the role of users to estimate the UX KPI for each application. The distribution of the labels averaged over the four experts (Fig. 2) shows a few outliers, but generally, according to the original dataset, the set of requirements has high UX scores with the first four UX labels: 'supportive,' 'easy,' 'efficient,' and 'clear. On the other hand, it shows that 'interesting' and 'exciting' labels have moderate scales. In contrast, the requirements in the benchmark dataset seem less 'inventive' and not up to the 'leading-edge.' The dataset can be downloaded from Kaggle[2].

The word cloud of this dataset is shown in Fig. 3. The word cloud was generated without considering any filtering or stop words removal. The figure shows that most requirements use the keywords "system", and "product", which indicates that such words are not discriminating the UX such as others ("allow", "display"). However, we cannot generalize these findings unless features (especially contextual ones) are plugged-in a proper machine learning model. The nature of the requirements datasets complicate machine learning models and needs further analysis to provide proper utility models.

Many authors use Fleiss' kappa to calculate the inter-annotator agreement; however, we opt not to calculate the inter-annotator agreement as it could provide inconsistency [34] for the following reasons: (1) each label has a scale between 1-7 for 969 records, (2) we have eight labels each with an ordered set of values, and (3) the order of values have a meaning, where for example, a class label of 2 is less than the same class label with 5. Therefore, Cronbach Alpha [35] was used to testify the internal consistency of each expert's scores and the root-mean-square error (RMSE) to see the deviation of scores from the means as we have a large sample size [36].

The Cronbach Alpha scores are shown in TABLE IV. The table shows an acceptable average reliability statistic except for the fourth expert. Therefore, the scores of expert four were eliminated from the benchmark dataset.

Moreover, we use the RMSE to compare paired scores (between any two experts), as shown in TABLE V. The table shows the averaged RMSE between paired experts averaged over the whole dataset. First, the RMSE was calculated per individual requirement item, where actual values and observed values are those coming from the paired experts. Furthermore, RMSE was calculated between each expert and the average scores for the first three experts and the average score for the four experts. The *avg3* (in TABLE V) is calculated by finding the average scores for the first three experts and then calculating the RMSE between each expert's actual and average scores. Similarly, *avg4* (in TABLE V) is calculated by averaging scores for the four experts and then finding the RMSE between the expert's actual value and the average score.

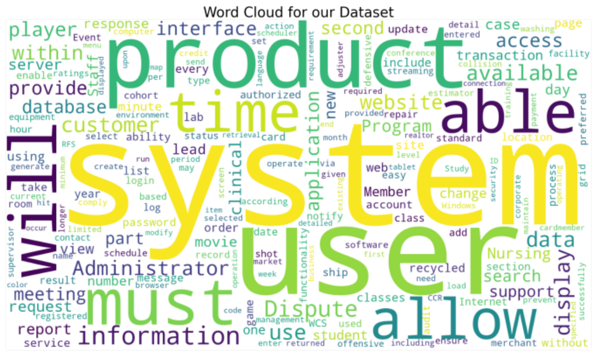

Fig. 2. Word Count of the Benchmark Dataset.

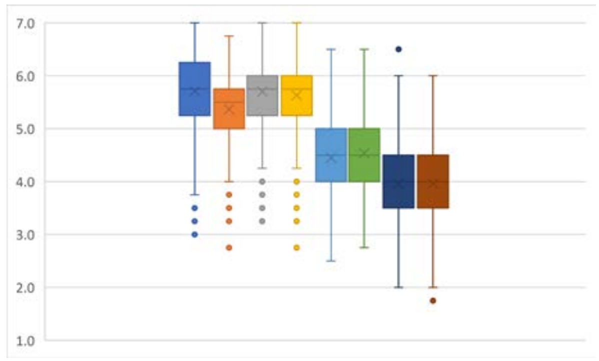

Fig. 3. Box and Whisker of the Average Scores.

TABLE IV. Cronbach Alpha Reliability for each Expert

	Average
Expert 1	0.69
Expert 2	0.70
Expert 3	0.85
Expert 4	(0.11) Omitted

The results of RMSE show that the fourth expert's scores deviate from the average scores (by a magnitude of approximately 2) between the other three experts; therefore, the score of the fourth expert was omitted from the benchmark dataset. According to TABLE V, the averaged RMSE is low and shows that results are consistent between experts, indicating that the dataset is acceptable for machine learning. The characteristics of the dataset are consistent with the literature that a suitable error estimation method is a method that reduces the dataset noise and scales out the absolute error to a minimum [38].

[2]https://bit.ly/3mLR9Cv

TABLE V. ROOT MEAN SQUARE ERRORS BETWEEN EXPERTS

	Exp1	Exp2	Exp3	Exp4	Av3*	Av4*
Exp1	0.00	2.90	4.68	4.38	0.80	1.03
Exp2	2.90	0.00	5.52	5.35	0.97	2.06
Exp3	4.68	5.52	0.00	3.01	0.77	1.06
Exp4	4.38	5.35	3.01	0.00	2.83	1.34
* av3: average(3) is calculated by finding the average scores for the first three experts and then calculating the RMSE between each expert and that score. Similarly, av4 (average 4) is calculated by averaging scores for the four experts and then finding the RMSE between the expert and the average score						

B. UX Evaluation

Given the dataset, the machine learning developer could tackle the problem as a multilabel or multi-regression problem. If the four annotations are averaged, then it could be seen as a multi-regression problem. However, if the resultant annotations were rounded to the original UX-scale(1-7), one might consider the problem a classification problem.

One approach to validating the benchmark dataset's output is experimenting with the dataset with different machine learning models. We have done a simple experiment on the first expert dataset using support vector machine (SVM), eXtreme Gradient Boosting (XGBoost), Decision Trees, and Bagging Classifier (KNN). We tackle machine learning problems as multilabel (8 labels) and multiclass (7 classes) regression problems. Nevertheless, predicted label-class values were rounded to the nearest label scale (1 to 7), making the comparison meaningful over precision, recall, and F1-score. First, stop words, and special characters were removed from the requirement text column, generating a sequence of words using the Tensorflow Tokenizer. Next, the default implementation of SVM and Decision Trees were used from the sklearn library and the latest python API for XGBoost.

The results are shown in TABLE VI. The results show that the XGB algorithm provides a reasonably acceptable F1-score (0.864), indicating that the benchmark dataset is helpful and applicable in UX predictive models. Decision trees were next performing machine learning model with an F1-score of 0.861. However, the SVM with one versus rest voting and the KNN Bagging (decreasing variance) was not performing well. Results could be due to the nature of SVM that would not work well with the multilabel problem, while the decision trees were giving an acceptable performance as it could build deep trees based on dataset instance and the 8 label values.

TABLE VI. RESULTS OF CLASSIFICATION METHODS

Method	Recall	Precision	F1-score
SVM	0.662	0.680	0.658
XGBoost	**0.864**	**0.863**	**0.864**
Decision Trees	0.861	0.861	0.861
Bagging(KNN)	0.747	0.746	0.746

In this experiment, the XGBoost classifier using the multiclass log loss function was a better choice for this dataset. Fig. 4 shows the ROC for the XGBoost model. The results show that class labels have a high area under the curve (except for label 3) where each score represents the average score of all eight labels. In other words, each average is calculated two times: the first time, averaging the prediction performance over a single label over the 7 class values (1 to 7), and then another average over the eight labels.

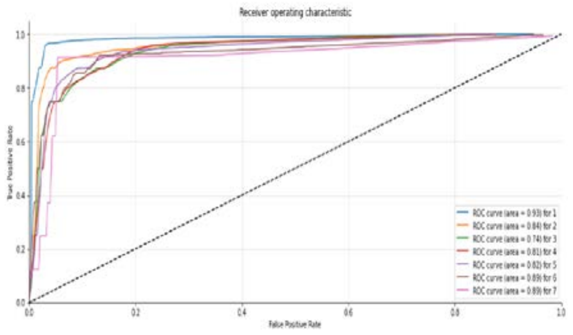

Fig. 4. ROC for XGBoost Model.

V. IMPLICATIONS AND LIMITATIONS

The new benchmark dataset allows the software engineers to predict the UX software value early in software development, and it allows the software engineers to generate UX-complaint software requirements. Moreover, the benchmark dataset extends the applicability of machine learning methods to measure a product UX instantly. The major advantage of the proposed machine learning models versus questionnaire-based approaches is automation, where UX could be evaluated on the fly at any point in time during software development. The UX evaluation could be used as a driving force for requirements analysis and validation [12]. Our study is consistent with the literature review conducted by Almeyda et al. [39] that showed that integrating the user experience and agile techniques are essential for requirements analysis.

The current dataset might have these limitations. It was assumed that all applications are similar in terms of UX features classification. Further research could enhance the dataset to support a customized dataset for each application. Moreover, the dataset is considered a small dataset; therefore, some deep learning models might not be a good choice.

VI. CONCLUSION

This paper annotated a dataset for UX based on textual requirements. The benchmark dataset helps software engineers predict the UX of an application before building a prototype or before releasing the software. Furthermore, the dataset could help software engineers generate UX-complaint requirements and as per the customer requirements. The dataset was tested with RMSE and some machine learning models. The results showed a low RMSE value between experts and a high F1-score for the XGBoost classifier. As a result, the dataset is considered the first of its kind to automate the UX evaluation. The dataset will be further expanded with new requirements, and the number of UX experts will be increased in the future. Moreover, the current contribution of this paper allows the research community to tackle requirements engineering issues using the current dataset by integrating the UX evaluation with the requirements engineering process.

REFERENCES

[1] A. G. Mirnig, A. Meschtscherjakov, D. Wurhofer, T. Meneweger, and M. Tscheligi, "A formal analysis of the ISO 9241-210 definition of user experience," in Proceedings of the 33rd annual ACM conference extended abstracts on human factors in computing systems, 2015, pp. 437–450.

[2] D. Quiñones and C. Rusu, "Applying a methodology to develop user eXperience heuristics," Computer Standards and Interfaces, vol. 66, no. January, p. 103345, 2019, doi: 10.1016/j.csi.2019.04.004.

[3] J. Nielsen, "The definition of user experience (UX). Nielsen Norman Group." Obtenido de https://www. nngroup. com/articles/definition-user-experience, 2018.

[4] Q. Zhang, "Mobile Internet Product Usage Scenarios and User Experience Design," in International Conference on Cognitive based Information Processing and Applications (CIPA 2021), 2022, pp. 857–865.

[5] P. Morville, "User experience design," Ann Arbor: Semantic Studios LLC, vol. 6, no. 2, 2004.

[6] A. Hinderks, M. Schrepp, F. J. D. Mayo, M. J. Escalona, and J. Thomaschewski, "Developing a UX KPI based on the user experience questionnaire," Computer Standards & Interfaces, vol. 65, pp. 38–44, 2019.

[7] K. Ohashi et al., "Focusing Requirements Elicitation by Using a UX Measurement Method," in IEEE 26th International Requirements Engineering Conference (RE), 2018, pp. 347–357.

[8] J. Nielsen, "How to conduct a heuristic evaluation," Nielsen Norman Group, vol. 1, pp. 1–8, 1995.

[9] A. Yeratziotis and P. Zaphiris, "A Heuristic Evaluation for Deaf Web User Experience (HE4DWUX)," International Journal of Human-Computer Interaction, vol. 34, no. 3, pp. 195–217, 2018, doi: 10.1080/10447318.2017.1339940.

[10] P. Kashfi, R. Feldt, A. Nilsson, and R. B. Svensson, "A conceptual ux-aware model of requirements," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 9856 LNCS, pp. 234–245, 2016, doi: 10.1007/978-3-319-44902-9_15.

[11] P. M. Fernandes, M. Lopes, and R. Prada, "Agents for automated user experience testing," Proceedings - 2021 IEEE 14th International Conference on Software Testing, Verification and Validation Workshops, ICSTW 2021, pp. 247–253, 2021, doi: 10.1109/ICSTW52544.2021.00049.

[12] I. Atoum et al., "Challenges of Software Requirements Quality Assurance and Validation: A Systematic Literature Review," IEEE Access, vol. 9, pp. 1–22, 2021, doi: 10.1109/ACCESS.2021.3117989.

[13] C. Ardito, P. Buono, D. Caivano, M. F. Costabile, and R. Lanzilotti, "Investigating and promoting UX practice in industry: An experimental study," International Journal of Human Computer Studies, vol. 72, no. 6, pp. 542–551, Jun. 2014, doi: 10.1016/j.ijhcs.2013.10.004.

[14] J. Choma, L. A. M. Zaina, and D. Beraldo, "UserX story: Incorporating UX aspects into user stories elaboration," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 9731, pp. 131–140, 2016, doi: 10.1007/978-3-319-39510-4_13.

[15] M. Schrepp, A. Hinderks, and J. Thomaschewski, "Construction of a Benchmark for the User Experience Questionnaire (UEQ)," IJIMAI, vol. 4, no. 4, pp. 40–44, 2017.

[16] B. Laugwitz, T. Held, and M. Schrepp, "Construction and evaluation of a user experience questionnaire," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2008, vol. 5298 LNCS, pp. 63–76, doi: 10.1007/978-3-540-89350-9_6.

[17] M. Schrepp, A. Hinderks, and J. Thomaschewski, "Applying the user experience questionnaire (UEQ) in different evaluation scenarios," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2014, vol. 8517 LNCS, no. PART 1, pp. 383–392, doi: 10.1007/978-3-319-07668-3_37.

[18] J. Cleland-Huang, S. Mazrouee, H. Liguo, and D. Port, "nfr." Zenodo, Mar. 2007, doi: 10.5281/zenodo.268542.

[19] M. Lima, V. Valle, E. Costa, F. Lira, and B. Gadelha, "Software engineering repositories: Expanding the PROMISE database," ACM International Conference Proceeding Series, pp. 427–436, 2019, doi: 10.1145/3350768.3350776.

[20] D. Biduski, E. A. Bellei, J. P. M. Rodriguez, L. A. M. Zaina, and A. C. B. De Marchi, "Assessing long-term user experience on a mobile health application through an in-app embedded conversation-based questionnaire," Computers in Human Behavior, vol. 104, 2020, doi: 10.1016/j.chb.2019.106169.

[21] P. H. Leong, O. S. Goh, Y. J. Kumar, Y. H. Sam, and C. W. Fong, "The Evaluation of User Experience Testing for Retrieval-based Model and Deep Learning Conversational Agent," International Journal of Advanced Computer Science and Applications, vol. 12, no. 4, pp. 216–221, 2021, doi: 10.14569/IJACSA.2021.0120429.

[22] K. Chrysafiadi and M. Virvou, "Evaluating the user experience of a fuzzy-based Intelligent Tutoring System," in 2021 12th International Conference on Information, Intelligence, Systems Applications (IISA), 2021, pp. 1–7, doi: 10.1109/IISA52424.2021.9555516.

[23] A. Krouska, C. Troussas, and C. Sgouropoulou, "A Cognitive Diagnostic Module Based on the Repair Theory for a Personalized User Experience in E-Learning Software," Computers, vol. 10, no. 11, 2021, doi: 10.3390/computers10110140.

[24] J. Hussain et al., "A multimodal deep log-based user experience (UX) platform for UX evaluation," Sensors (Switzerland), vol. 18, no. 5, 2018, doi: 10.3390/s18051622.

[25] K. Koonsanit and N. Nishiuchi, "Classification of user satisfaction using facial expression recognition and machine learning," IEEE Region 10 Annual International Conference, Proceedings/TENCON, vol. 2020-Novem, pp. 561–566, 2020, doi: 10.1109/TENCON50793.2020. 9293912.

[26] C. I. Nwakanma, M. S. Hossain, J.-M. Lee, and D.-S. Kim, "Towards Machine Learning Based Analysis of Quality of User Experience (QoUE)," International Journal of Machine Learning and Computing, vol. 10, no. 6, pp. 752–758, 2020, doi: 10.18178/ijmlc.2020.10.6.1001.

[27] Y. Li and C. Liu, "User Experience Research and Analysis Based on Usability Testing Methods," in Lecture Notes in Electrical Engineering, 2021, vol. 754 LNEE, pp. 263–267, doi: 10.1007/978-981-16-0503-1_39.

[28] M. Eriksson and B. Ferwerda, "Towards a User Experience Framework for Business Intelligence," Journal of Computer Information Systems, vol. 61, no. 5, pp. 428–437, 2021, doi: 10.1080/08874417.2019.1693936.

[29] H. Jang and S. H. Han, "User experience framework for understanding user experience in blockchain services," International Journal of Human-Computer Studies, vol. 158, p. 102733, 2022, doi: https://doi.org/10.1016/j.ijhcs.2021.102733.

[30] A. Casare, T. Basso, and R. Moraes, "User Experience and Trustworthiness Measurement: Challenges in the Context of e-Commerce Applications," in Proceedings of the Future Technologies Conference (FTC) 2021, Volume 1, 2022, pp. 173–192.

[31] I. Atoum, "A Scalable Operational Framework for Requirements Validation Using Semantic and Functional Models," ACM International Conference Proceeding Series, pp. 1–6, 2019, doi: 10.1145/3305160.3305166.

[32] I. Atoum, "Requirements Elicitation Approach for Cyber Security Systems," i-manager's Journal on Software Engineering, vol. 10, no. 3, pp. 1–5, 2016, doi: doi.org/10.26634/jse.10.3.4898.

[33] J. D. Brown, "The Cronbach alpha reliability estimate," JALT Testing & Evaluation SIG Newsletter, vol. 6, no. 1, 2002.

[34] R. Falotico and P. Quatto, "Fleiss' kappa statistic without paradoxes," Quality & Quantity, vol. 49, no. 2, pp. 463–470, 2015, doi: 10.1007/s11135-014-0003-1.

[35] D. G. Bonett and T. A. Wright, "Cronbach's alpha reliability: Interval estimation, hypothesis testing, and sample size planning," Journal of organizational behavior, vol. 36, no. 1, pp. 3–15, 2015.

[36] T. Chai and R. R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)?--Arguments against avoiding RMSE in the literature," Geoscientific model development, vol. 7, no. 3, pp. 1247–1250, 2014.

[37] A. Al-Hunaiyyan, R. Alhajri, B. Alghannam, and A. Al-Shaher, “Student Information System: Investigating User Experience (UX),” International Journal of Advanced Computer Science and Applications, vol. 12, no. 2, pp. 80–87, 2021, doi: 10.14569/IJACSA.2021.0120210.

[38] I. Atoum and M. R. Ayyagari, “Effective semantic text similarity metric using normalized root mean scaled square error,” Journal of Theoretical and Applied Information Technology, vol. 97, no. 12, pp. 3436–3447, 2019.

[39] S. Almeyda, C. Zapata Del Río, and D. Cohn, “Integration of User Experience and Agile Techniques for Requirements Analysis: A Systematic Review,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2021, vol. 12779 LNCS, pp. 187–203, doi: 10.1007/978-3-030-78221-4_13.