Journal on Interactive Systems, 2025, 16:1, doi: 10.5753/jis.2025.4099

UX-MAPPER: An automated approach to analyze app store reviews with a focus on UX

Walter T. Nakamura [**Federal University of Technology - Paraná (UTFPR), Federal University of Amazonas (UFAM)** | ***waltertakashi@utfpr.edu.br***]
Edson C. C. de Oliveira [**Amazonas State Treasury Department (SEFAZ/AM)** | ***edsono@gmail.com***]
Elaine H. T. de Oliveira [**Federal University of Amazonas (UFAM)** | ***elaine@icomp.ufam.edu.br***]
Tayana Conte [**Federal University of Amazonas (UFAM)** | ***tayana@icomp.ufam.edu.br***]

Academic Department of Computing (DACOM), Federal University of Technology - Paraná (UTFPR), Rua Rosalina Maria dos Santos, 1233, Vila Carolo, Campo Mourão, PR, 87301-899, Brazil.

Received: 06 February 2024 • **Accepted:** 18 October 2024 • **Published:** 01 January 2025

Abstract: The mobile app market has increased substantially in the past decades, and the myriad options in the app stores have made users less tolerant of low-quality apps. In this competitive scenario, User eXperience (UX) has emerged as an essential factor in standing out from competitors. By understanding what factors affect UX, practitioners could focus on factors that lead to positive UX while mitigating those that affect UX negatively. In this context, app store reviews emerged as a valuable resource for investigating these influential factors. However, analyzing millions of reviews can be costly and time-consuming. This article introduces UX-MAPPER, a tool designed to analyze app store reviews and assist practitioners in pinpointing factors that impact UX. We applied the Design Science Research method to develop UX-MAPPER iteratively and rooted in a robust theoretical background. We performed exploratory studies to investigate the problem, a systematic mapping study to identify UX-affecting factors, and an empirical study to ascertain practitioners' relevance and acceptance of UX-MAPPER. In general, the participants recognized the relevance and utility of UX-MAPPER in enhancing the quality of existing apps and exploring reviews of competing apps to identify user preferences, requests, and critiques regarding functionalities and features. However, the output quality requires refinement to better convey the benefits of the results, especially for practitioners with prior experience with automated approaches. From the participants' feedback, we defined a set of suggestions to extract more useful features, which can contribute to future studies involving user review analysis. Based on the results of this research, we present the contributions to the area of HCI and possible developments for future research.

Keywords: User experience, User reviews, Machine learning, App stores

1 Introduction

The global mobile device market has grown exponentially in recent decades. This popularization has resulted in the development of thousands of applications created by small and large companies to cater to various audiences and distributed through app stores, reaching the mark of 11.8 million apps in 2020[1]. With a wide variety of apps available, users have developed a low tolerance for faulty or low-quality applications, leading them to quickly remove and replace such apps [Durelli *et al.*, 2018]. In this competitive scenario, companies have increasingly focused on design and user experience to create unique, satisfying, and enlightening experiences that secure a place on users' devices [Alves *et al.*, 2014]. Understanding the factors (i.e., every aspect related to the application or the user associated with a positive, negative, or neutral perception of the experience) influencing users' perceptions of UX has become essential for maintaining a competitive edge.

Recent studies have highlighted key factors that can significantly impact UX evaluations. Notably, users have sometimes rated their UX as positive despite encountering interaction problems [Nakamura *et al.*, 2019a]. For instance, even users who experienced negative emotions during their interactions still rated their UX positively when reflecting on their experience through a questionnaire [de Andrade Cardieri and Zaina, 2018]. In digital games, a study found that sadness was the most frequently mentioned emotion, yet players often found these experiences rewarding, giving high ratings for appreciation and enjoyment [Bopp *et al.*, 2016].

These findings have important implications for users, practitioners, and researchers. Identifying the factors influencing UX can help practitioners focus on enhancing positive aspects while mitigating negative ones. Users would benefit from products that better meet their needs and provide more positive experiences. For practitioners, understanding these factors could prevent unnecessary efforts in developing features or fixing issues that have minimal impact on UX. It could also guide researchers to conduct studies with fewer biases, acknowledge these factors beforehand, and take action to reduce their effects. For instance, negative emotions like sadness, often seen as undesirable in software interactions, have played a significant role in user perception, particularly in gaming [Bopp *et al.*, 2016]. Users may value intense emotional experiences, even negative ones, which can lead to higher ratings of appreciation and enjoyment. This suggests that the impact of UX factors can vary depending on the type of software and other variables, such as gender

[1] https://www.riskiq.com/wp-content/uploads/2021/01/RiskIQ-2020-Mobile-App-Threat-Landscape-Report.pdf

and culture. By identifying these factors, it becomes possible to develop more accurate UX evaluation methods and create guidelines that focus on the most influential aspects of the user experience.

One way to identify these factors is through UX evaluations. Although various UX evaluation methods have been proposed in the last decade [Rivero and Conte, 2017], they are often costly and time-consuming, requiring highly trained personnel and numerous users to perform tasks, which may not be feasible in an agile development context. In this scenario, user reviews from app stores can serve as a valuable data source for extracting information that drives development efforts and improves future releases by identifying requirements, improvement requests, and bugs [Guzman and Maalej, 2014].

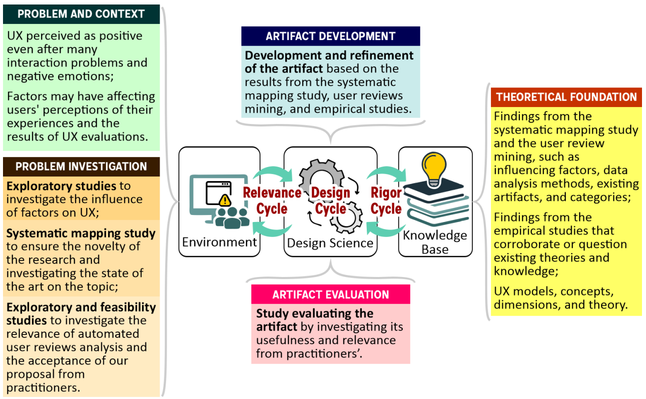
In contrast to the feedback collected from controlled experiments, app stores promote a favorable environment where users worldwide can express their opinions and experiences spontaneously, describing what they like or hate the most and helping developers identify which problems to solve and improvements to make [Santiago and Marques, 2023]. By identifying the factors affecting UX, it would be possible to: i) minimize bias in UX evaluations; ii) create techniques that guide developers into reliable results by taking into account the influence of these factors; iii) avoid rework in the app development process by considering the existence of these factors beforehand; iv) support the redesign of an app by identifying the impact of the factors affecting UX. Thus, this research aimed to answer the question: "*How can we identify the factors affecting users' perceptions of their experience in user reviews from app stores?*".

This article is an extended version of the paper initially published in [Nakamura *et al.*, 2024] and proposes an approach called UX-MAPPER (User eXperience Method to Analyze App Store Reviews). In this work, we performed two additional analyses to fill two research gaps not covered in the original paper: (i) How to provide more useful features for practitioners?, and (ii) What is the acceptance of UX-MAPPER according to practitioners' experience in analyzing user reviews and using automated approaches? Regarding the first question, we identified a set of characteristics practitioners reported when evaluating a feature's usefulness and developed a set of suggestions to extract more useful features, which can contribute to future studies involving user review analysis. Regarding the second question, we identified different improvement opportunities according to the profile of the participants. In addition to these two analyses, we provided details on developing and refining UX-MAPPER, presenting the classifiers' training process results and the rationale behind each decision. We also made available the labeled dataset used to train the classifiers, which could serve as the basis for developing other approaches that analyze user reviews with a focus on UX. The comprehensive analysis of the results regarding the additional constructs from the original TAM questionnaire [Davis, 1989] triangulated with the qualitative analysis can also serve as a reference for future studies involving technology acceptance by practitioners from the industry. With this work, we expect to support practitioners in the software development process by providing an approach to analyze app store reviews and identify the factors leading to positive or negative UX. From the researchers' perspective, the methodology we followed to develop UX-MAPPER can serve as the basis for developing new artifacts, especially tools and methods that evaluate UX.

The remainder of this article is structured as follows. Section 2 details the methodology followed in this research. Sections 3, 4, and 5 present our studies, in which the findings served as the basis for developing UX-MAPPER. Section 3 presents an exploratory study to investigate some influencing factors from the literature. In Section 4, we present the results from a systematic literature mapping addressing factors that affect UX from publications that analyzed reviews from app stores. In Section 5, we describe the results of two studies conducted with practitioners from the industry to investigate the feasibility of creating an automated approach that analyzes app store reviews. Section 6 details the development of UX-MAPPER based on the results presented in Sections 3, 4, and 5. Section 7 presents the results of the evaluation of UX-MAPPER by practitioners from the industry. Section 8 presents related work. Section 9 presents the threats to validity. Finally, Section 10 concludes the article by presenting the main contributions of this research and future work.

2 Research Methodology

We applied the Design Science Research (DSR) in this work. DSR is a research method consisting of an iterative process that aims to design and investigate innovative artifacts [Wieringa, 2014], contributing with new knowledge to the body of scientific evidence [Hevner and Chatterjee, 2010]. In DSR, the artifact is improved iteratively to solve a problem and comprises three cycles: relevance, design, and rigor [Hevner and Chatterjee, 2010]. Figure 1 presents an overview of the method. We present the concept behind each cycle and an overview of the steps performed in each cycle below.



UX perceived as positive even after many interaction problems and negative emotions;

Factors may have affecting users' perceptions of their experiences and the results of UX evaluations.

Exploratory studies to investigate the influence of factors on UX;

Systematic mapping study to ensure the novelty of the research and investigating the state of the art on the topic;

Exploratory and feasibility studies to investigate the relevance of automated user reviews analysis and the acceptance of our proposal from practitioners.

Development and refinement of the artifact based on the results from the systematic mapping study, user reviews mining, and empirical studies.

Study evaluating the artifact by investigating its usefulness and relevance from practitioners'.

Findings from the systematic mapping study and the user review mining, such as influencing factors, data analysis methods, existing artifacts, and categories;

Findings from the empirical studies that corroborate or question existing theories and knowledge;

UX models, concepts, dimensions, and theory.

Figure 1. Overview of the DSR cycles employed in this research.

Research opportunities and problems in a given application environment are identified in the **relevance cycle**. The environment in Figure 1 refers to where the phenomenon of interest (i.e., the problem) is observed and where the artifact operates. In this cycle, the researcher verifies whether the proposed artifact improves the environment, how these improvements can be measured, and whether additional iterations in the relevance cycle will be necessary [Hevner and Chatterjee, 2010].

In our previous studies [Nakamura *et al.*, 2019b, 2020], we realized that many users still evaluated their UX as positive, even when facing interaction problems during the tasks. From this previous experience, we performed an initial adhoc literature review [Nakamura *et al.*, 2019a] to search for studies that reported similar findings and identify research gaps. We hypothesized that there should be factors that weigh more in the users' perception of the experience, leading to contradictory results. Thus, we began investigating the effect of factors on UX by carrying out an empirical study (section 3). The findings supported our initial hypothesis, indicating that factors can affect how users perceive their experience. Such findings highlighted that the problem is real and worth investigating.

To investigate what is known in the literature about these factors and assess the novelty of our research, we performed a systematic mapping study to address publications that analyzed user reviews from app stores (section 4). Our focus on app store reviews is because they are considered the "voice of users" [Guzman and Maalej, 2014], from which practitioners could obtain information to improve the quality of their app. The broad view of a systematic mapping study allows gathering results from several studies in various datasets and contexts to obtain a more thorough analysis and draw conclusions that would be hard to get through isolated app review studies.

After identifying the factors affecting UX, we conducted an exploratory study to investigate the relevance of automating user review analysis from the practitioners' points of view (section 5.1). Based on the findings from this study, we developed an initial proposal and evaluated its acceptance through a feasibility study (section 5.2). The results highlighted the relevance of our proposal and the main features that should be implemented in our artifact.

The **rigor cycle** consists of identifying state of the art to develop an artifact with a solid theoretical foundation. Existing artifacts, processes, experiences, and expertise that define the state-of-the-art in the research application domain are identified [Hevner and Chatterjee, 2010]. This cycle also adds to the knowledge base, such as extensions to original theories and methods, new meta-artifacts, such as design products and processes, and all the experiences gained by employing the artifact in the application environment [Hevner, 2007].

In this research, the development of the artifact is grounded on theoretical foundations from different sources. Our first source is related to the UX theory, which involves models, concepts, measures, and dimensions defined by previous works in the literature (e.g., [Hassenzahl, 2007; Law *et al.*, 2014]). The second source is the findings from our systematic mapping study. Finally, we have the experience and results from each empirical study we conducted to test hypotheses and derive conclusions that support and guide the development of the artifact.

The **design cycle** is the heart of the DSR project and consists of developing the artifact based on the theoretical foundation, knowledge, and previous experiences obtained in the rigor cycle [Hevner, 2007]. The artifact is also evaluated through its application in the environment. The results allow for identifying improvement opportunities for the next cycle until a satisfactory design is achieved [Hevner and Chatterjee, 2010].

We developed our artifact grounded on the findings obtained in the previous cycles and refined our artifact iteratively (section 6). To do so, we evaluated different Machine Learning (ML) approaches from the literature and employed widely known technologies to support our artifact. After developing the tool, we validated it by conducting a study with practitioners to investigate its relevance and usefulness in the software development context (section 7).

It is noteworthy that this research project was approved by the ethics committee of UFAM - Certificate of Presentation for Ethical Consideration–CAAE number 40928120.6.0000.5020. All participants signed the informed consent form, which explained the purpose of the research, voluntary participation, treatments for possible risks and discomforts, confidentiality of data, and the possibility of withdrawing from the study at any time. All information collected in the studies was treated as confidential. We did not collect any personal information that could identify the participant, and the answered questionnaires were all anonymized and destroyed after being transcribed. Audio and video recordings were made only with the participant's permission and destroyed after transcription.

3 Investigating the influence of factors on UX

Previous studies have explored factors such as the number of problems [Nakamura *et al.*, 2020], previous experience [Sagnier *et al.*, 2020], and interaction sequencing [Cockburn *et al.*, 2017]. However, some gaps remain open, requiring further studies. In our previous work [Nakamura *et al.*, 2020], we found that inspectors evaluated the UX of a product lower than users did, possibly due to the number of problems revealed during the inspection process. However, the users' profiles may also have contributed to this difference, given that they only used computers occasionally, thus having low experience compared to inspectors. The impact of previous experience on UX, particularly with novel interaction methods, is still unclear and requires further investigation. While Cockburn *et al.* [2017] observed a significant impact of interaction sequencing, Gutwin *et al.* [2016] reported mixed results depending on the type of game, suggesting that this factor might not always significantly affect UX. Thus, it is necessary to investigate its impact on UX, especially in the context of mobile apps.

In this study, we investigated the effect of the factors above: number of problems, interaction sequencing, and prior experience. To do so, we evaluated an app designed to facilitate shopping in local markets by adopting a novel interaction approach using a chatbot. We compared the UX from both inspectors' and users' points of view to investigate whether the number of problems identified during the inspection and user testing influences participants' perception of the experience. We also evaluated the effect of interaction sequencing in an actual mobile application by manipulating tasks with different levels of effort. Finally, we investigated the effect of prior experience by evaluating a novel shopping application that uses a chatbot, changing interaction paradigms.

In this study, we applied three evaluation methods. For inspection, we selected UX-Tips [Marques *et al.*, 2021], a heuristic-based method that evaluates a set of factors, such as aesthetics, emotion, and engagement. For testing, we adopted the Think-Aloud method. Finally, we adopted the shortened version of the User Experience Questionnaire (UEQ) [Schrepp *et al.*, 2017] for the UX evaluation to reduce the time required for the study. We also added the Valence dimension of the Self-Assessment Manikin method [Bradley and Lang, 1994] to assess participants' overall satisfaction.

We built two scripts to investigate the effect of interaction sequencing: one for the negative beginning and positive ending condition (+end) and another for the positive beginning and negative ending condition (-end). Each participant was randomly assigned to only one condition. At the end of the evaluation, the participants filled in the UEQ [Schrepp *et al.*, 2017]. We also asked those who had already used similar apps to rate the UX of a similar application they remembered before evaluating the target application to better understand the relationship between previous and current experiences.

The results indicated that the number of problems and prior experience affect UX. The number of problems mainly affected the PQ dimension but not the HQ. Inspectors perceived the PQ dimension significantly more negatively than users. We also found a strong and moderate negative correlation between the number of problems with the PQ dimension and satisfaction, respectively. As inspectors are focused on identifying problems, it might have affected their perception of the app, indicating that the method can significantly influence the results of UX evaluations.

Regarding the interaction sequencing factor, generally, participants from the -end group provided lower ratings than the +end group. However, we did not find a significant difference between the two conditions.

Finally, regarding the prior experience with similar shopping applications, we only analyzed the data from the testing group, given that only one participant from the inspection group did not have previous experience with this type of application. Both groups evaluated the HQ of the application positively. In turn, we found a significant difference in PQ between participants with prior experience, who had neutral perception, and participants without prior experience, who perceived it positively.

The correlation analysis revealed that users' overall satisfaction without prior experience is strongly associated with hedonic aspects. The more innovative and interesting the application is, the greater the users' satisfaction with it. In turn, for users with prior experience with similar applications, both pragmatic and hedonic aspects play an important role in their satisfaction, with a stronger emphasis on the former.

The results support our initial hypothesis on the influence of factors on UX evaluation and their impact. Although the participants faced many problems, they still perceived the UX positively, especially those without experience with similar apps. Such findings indicate that previous experience weighs on UX evaluations and affects users' overall perception of the experience. This highlighted the relevance of investigating such factors to assess their impact on UX.

4 Systematic Literature Mapping

After our initial findings presented in the previous section, we proceeded to the second iteration of the relevance cycle, where we assessed the novelty of our research and identified potential gaps to be explored by conducting a systematic mapping study to identify factors that can affect UX. The knowledge obtained from this iteration also led to a first iteration over the rigor cycle by contributing to building a body of knowledge on the topic.

As our goal was to investigate UX-related factors in different types of products, we focused on publications that analyzed user reviews from app stores. In this systematic mapping, we aimed to answer the following research question: "*What are the UX-related factors that influence users' evaluations in app store reviews, and how do they affect UX?*".

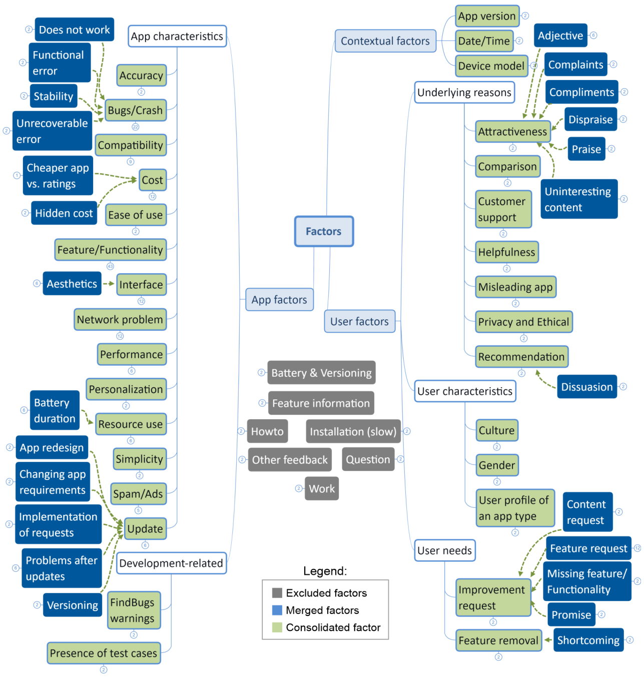


Figure 2. Factors mapping and merging process.

From 25 publications accepted, we identified 31 unique UX-related factors. We defined three high-level conceptual categories to group the factors according to the definition of UX. *App Factors* are related to the app's characteristics, functionalities, features, and development. *User factors* are related to users, such as their profile, needs, and the reasons for their positive or negative evaluations. *Context Factors* comprise factors related to the environment where the interaction occurred. Next, we refined the set of factors by analyzing the description of each factor and grouping them according to their concept (see Figure 2).

This systematic mapping study revealed a varied effect of these factors. We found that negative reviews are prevalent in factors related to features and functionality issues (e.g., *Performance, Feature Removal, Compatibility, Net-*

work Problem). On the other hand, positive reviews tend to describe overall qualities and aspects of the app, emphasizing factors related to general perceptions and human aspects (e.g., *Helpfulness, Customer Support, Ease of Use, Culture*).

Certain factors have different impacts depending on their polarity. Negative reviews regarding the app's cost and interface can decrease ratings, while positive reviews have little effect on the overall rating. Additionally, specific factors are more commonly mentioned in certain types of apps. For mobile games, *Attractiveness, Stability,* and *Cost* were identified as the top factors. *Privacy and Ethical* had the greatest negative impact on UX, while *Spam/Ads* was the most critical factor in a mobile game, leading to the lowest ratings. The *Update* factor showed varying effects. While minor improvements users request can boost ratings, a complete interface redesign can result in dissatisfaction. Negative evaluations often stem from usability issues, update problems, and broken functionalities caused by new releases. Therefore, developers must be cautious when updating their apps and pay close attention to the reviews, particularly after releasing an update.

From a practitioner's perspective, our findings provide insights into factors to consider when developing or improving mobile apps focusing on UX. From an academic perspective, researchers could assign different weights to these factors when evaluating UX. They could also propose approaches that automatically analyze user reviews to identify influential factors and their impact on the app developed. The findings from this study served as a basis for defining the factors that UX-MAPPER should consider when analyzing user reviews. Also, they highlighted the importance of analyzing user reviews automatically to identify which factor to prioritize, given that their effect varies according to the context.

5 Investigating Practitioners' Perceptions

This section presents our third iteration of the relevance cycle. We conducted an exploratory study to investigate how app store reviews are used in the software development industry and a feasibility study to assess the acceptance of our initial proposal.

5.1 Exploratory Study

This study aimed to understand how practitioners analyze user reviews from app stores, their importance in the software development process, and the challenges involved. We also investigated practitioners' opinions towards an automated approach to analyzing app store reviews. To do so, we conducted semi-structured interviews with three practitioners from distinct software development companies in Manaus (Brazil) working on projects developing mobile applications, selected by convenience.

The results indicated that the companies know the importance of user reviews for software development and evolution. They all analyzed user reviews to improve their software at some point in the project. Regarding the main challenges, two interviewees reported the lack of constructive information in the reviews and the time required to analyze them. Finally, they agreed that an automated approach would contribute much to their work, especially to speed up the development process.

The results revealed that the problem under study is relevant, and there is a need for approaches that automate the analysis and provide relevant information for the development team to improve the company's software. Such findings reinforced the importance of our proposal, which motivated us to assess its feasibility through an initial prototype.

5.2 Feasibility Study

This study aimed to answer the question: "*What is the feasibility of an automated tool that analyzes app store reviews to support identifying improvement opportunities from practitioners' perspective?*". To do so, we developed an initial prototype of a tool that analyzes user reviews and extracts the most frequent terms. We asked the participants to interact with the prototype, analyze the terms extracted and the reviews associated with them, and elicit requirements.

Considering the pandemic scenario, we focused on extracting reviews from technologies that support remote teaching. Among them, we selected Kahoot![2], one of the most popular game-based learning platforms. When the practitioner clicks on a term in the word cloud, the tool shows a list of reviews that contain this term.

We carried out this study with six practitioners from distinct companies that did not participate in the exploratory study. They all had previous experience with requirements elicitation, performing this task in at least one project, but only two had elicited requirements from app store reviews.

Due to COVID-19 restrictions, we carried out the study using the Google Meet[3] platform. We began introducing the study and its goals. Next, we asked the participants to sign the informed consent form and complete a characterization questionnaire. They were given a link to the tool and time to explore its functionalities. Then, we instructed the participants to think about requirements to improve the app or develop a concurrent application based on the reviews of the five most frequent terms. Finally, we asked the participants to fill out the post-study questionnaire.

The results highlighted the potential of an automated approach to analyzing app store reviews. Overall, the proposal was positively accepted. The participants found it easy to use and understand and agreed they would use the tool if made available. However, improvements were needed. Some participants pointed out that some reviews required much interpretation to analyze due to the lack of context, while other reviews did not lead to requirements, especially the purely emotional ones. The participants also felt the need for filtering and sorting functionalities and a more precise graphical representation to compare the frequency of the terms.

To address these points, we decided to make the following improvements to develop UX-MAPPER: i) to implement state-of-the-art feature extraction approaches instead of only presenting the most frequent terms; ii) to classify the reviews

[2] https://kahoot.it/

[3] https://meet.google.com/

according to identified factors, making it easier for practitioners to find specific topics; we also included sorting and filtering functionalities based on the number of thumbs up and star ratings, as these metrics have been shown to indicate helpful reviews [Palomba *et al.*, 2017]; iii) to present the top features using bar charts to facilitating comparison of their frequencies. In the next section, we detail the development of our proposal.

6 UX-MAPPER Development

This section presents our first iteration of the Design Cycle. The previous studies provided the theoretical background to begin developing our artifact. The main findings were as follows:

Findings from the first empirical study (section 3):

- *Different factors can affect UX evaluations:* We identified that some factors can significantly affect users' perception of their experiences. From this initial finding, we performed a systematic mapping study to investigate what is known in the literature regarding influencing factors.

Finding from the systematic mapping study (section 4):

- *Several factors can affect UX, and their effect varies according to the context:* The study resulted in a set of factors that served as input for defining which UX-MAPPER should consider when analyzing user reviews. It also highlighted the importance of developing an approach that automatically analyzes user reviews and allows for the identification of which factors and features to prioritize, given that their effect varies according to the context.

Findings from the exploratory and feasibility studies (section 5.1 and 5.2):

- *Practitioners consider user feedback from app stores in their work:* This finding strengthened the importance of our proposal for software development and evolution;
- *Identifying relevant reviews is time-consuming:* Our artifact should sort the reviews by relevance and group them into factors to facilitate finding reviews related to a given topic;
- *Word cloud is not the best way to present features:* The artifact should provide a graphical representation that makes the data comparison and analysis more intuitive, such as a bar chart ordered by the frequency of each feature to create a rank that allows comparing the features more precisely.

The following subsections describe the artifact architecture, its development and refinement process, and its functioning.

6.1 UX-MAPPER's Architecture

The tool comprises three components (see Figure 3): 1) Data Gathering and Processing; 2) Factor Extraction; and 3) Feature Extraction.

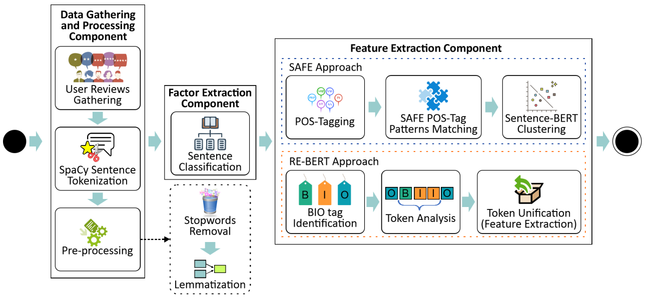


Figure 3. UX-MAPPER architecture.

The **Data Gathering and Processing Component** obtains user reviews from app stores and processes the data that the other components will use. First, it extracts user reviews through the Google-Play-Scrapper API for Python[4] and tokenizes them into sentences using SpaCy[5], a state-of-the-art NLP tool [Al Omran and Treude, 2017]. Finally, it performs preprocessing steps by removing stopwords and reducing different inflected forms of a word into their lemma (using SpaCy).

The **Factor Extraction Component** takes the output of the Data Gathering and Processing Component to analyze the data and label the sentences according to the factor identified. Each sentence can be labeled into more than one factor. The component uses the Support Vector Machine (SVM), a supervised classifier that has been proven to be highly effective on a variety of tasks, such as text classification, pattern recognition, and computer vision [Nalepa and Kawulok, 2018].

Finally, the **Feature Extraction Component** analyzes the reviews from each factor and extracts a set of terms that may be relevant for practitioners to improve the quality of their apps. We implemented two state-of-the-art approaches identified in the literature: SAFE (Simple Approach for Feature Extraction) [Johann *et al.*, 2017] and RE-BERT [de Araújo and Marcacini, 2021].

6.2 Factor Extraction Component

To develop this component, we began selecting the factors to be analyzed by UX-MAPPER. We analyzed each factor returned in the systematic mapping study and checked whether it could be addressed through user reviews and ratings. *Findbugs Warnings* and *Presence of test cases* relied on source code analysis. *Device model*, *Culture*, and *Gender* depended on data that were not publicly available. *User Profile of an App Type* requires a comparison of an entire app category, which can consume a lot of resources and processing time. *Feature/Functionality* vary according to each app and is usually mentioned in reviews from other factors, such as *Improvement request*, *Bugs/Crash*, and *Feature removal*, being possible to identify them, for instance, through collocation algorithms. Thus, we did not include these factors in UX-MAPPER. Due to overlapping issues identified during the pilot study (see the next subsection), we also merged the *Network Problem* into the *Bugs/Crash* factor.

[4] https://pypi.org/project/google-play-scraper/
[5] https://spacy.io/

6.2.1 Pilot Study

We first performed a pilot study before labeling a large set of reviews. To diversify our sample, we selected one app from five different categories: Entertainment (Netflix), Communication (WhatsApp), Tool (CCleaner), Social (TikTok), and Game (Garden Scapes). From each app, we extracted 10,000 reviews written in English. In this study, we selected 20 random reviews, which resulted in 51 sentences.

This pilot study involved four people: the main researcher of this work and three computer science undergraduate students. Each person performed the labeling process individually. Additionally, the three students had to discuss their classifications and reach a consensus to provide a single labeled set.

Six out of 51 sentences had disagreements. The main cause was the *Bugs/Crash* and *Network Problem* factors due to the difficulty in differentiating a connectivity problem from a bug in the app. Thus, we decided to merge the Connectivity factor into the *Bugs/Crash* factor, as the latter's definition is broader.

6.2.2 Model Training and Testing

We trained our model iteratively by making adjustments and decisions grounded on data. For all three iterations, we performed the following steps: manual sentence labeling, data preprocessing, model training, and model testing.

In the data preprocessing step, we made the text lowercase, removed stopwords, and applied lemmatization. To create our training set, we transformed each class (i.e., factor) into dummy-coded variables (e.g., 0- false, 1- true) using the MultiLabelBinarizer function from the scikit-learn[6] library. After transforming to a binary matrix, we extracted features with one and two words (n-grams = 1,2) from the sentences using the CountVectorizer function, which converts text documents into a matrix of token counts. We also tested with TF-IDF (Term Frequency-Inverse Document Frequency), which combines the frequency of the term with the inverse document frequency to calculate its importance in the document [Maalej *et al.*, 2016].

We trained our model by employing four classifiers commonly applied in the field of user reviews mining, providing good results [Bakiu and Guzman, 2017; McIlroy *et al.*, 2015; Panichella *et al.*, 2015]: J48, Logistic Regression, Linear SVC (SVM), and Multinominal Naïve-Bayes. Given that these classifiers are of the binary type and our problem is multilabel, we used the OneVsRestClassifier algorithm from scikit-learn to make the training and testing process possible. All the processing was performed in a notebook equipped with an Intel Core i7-8565U processor, 8GB DDR4, NVIDIA GeForce MX110 2GB DDR5, and Corsair SSD MP510 480GB.

To minimize the bias of random sampling of the training set, we performed a 10-fold cross-validation. Given that our dataset comprised multi-labeled instances with imbalanced classes, we applied the Iterative Stratification algorithm [Sechidis *et al.*, 2011]. This algorithm distributes the positive examples of each class into each fold to reduce the chance of obtaining folds without positive examples, which could affect the results. In each iteration, we calculated the *micro* and *macro* metrics for Precision, Recall, and F1 score. *Micro-averaging* aggregates the basic four quantities (True-Positives, False-Positives, True-Negatives, and False-Negatives) to be treated as a unique metric to calculate each binary classification metric. *Macro-averaging* is the sum of the result of the binary classification metric from all classes divided by the number of classes.

6.2.3 First Iteration

When labeling the sentences in the first iteration, we realized that reviews from the Game category have specificities that make it hard for the model to learn. Users report many specific problems related to a given stage/phase of the game with various narratives that do not use common words that indicate a bug or a problem, making it challenging to identify a pattern. Some terms used in this category can also have a different meaning. The words "performance" and "slow" may not be related to how fast the application runs but to the gamer's progress and how the story evolves. Due to this specificity of games, we decided to remove it from the analysis. At the end of this initial labeling process, we labeled 532 reviews and 1,399 sentences. Among them, 733 sentences were not associated with any factors, giving a total of 666 sentences labeled with at least one factor.

In this first iteration, the performance of the classifiers was very poor. The SVM, LR, and NB achieved high precision regarding micro-averaged metrics, with over 87% of the instances classified correctly. However, their recall was very low, indicating they missed many instances. J48, in turn, had poor precision, with around 60% of the instances classified correctly, but achieved greater recall. All four classifiers performed very poorly in all metrics regarding the macro-averaged measures. It is mainly due to the small sample size and imbalanced classes, given that the number of instances varied from 2 (Accuracy) to 233 (Bugs/Crash). Finally, the additional step to apply TF-IDF required a little more time for the model to fit (Time-to-fit - TTF). It also had few effects on classification, with a slight increase in precision at the expense of recall, which decreased F1-score.

When analyzing the wrong-labeled sentences, we identified two main issues: i) the NLTK tokenizer had problems splitting long and unstructured reviews (i.e., without proper punctuation) - longer sentences may have more factors associated, making it difficult for the classifier to learn the most important features of the class; ii) some sentences were labeled by considering their implicit meaning obtained through the interpretation of the labeler, something that the algorithm could not identify.

6.2.4 Second Iteration

In this second iteration, we tested three tokenization libraries: TextBlob, Stanford CoreNLP, and Spacy. Among the four selected libraries, SpaCy obtained the best results, being capable of splitting long reviews that do not have a period or other form of punctuation that indicates the end of a sentence. As the entire set had to be modified due to the different sen-

[6] https://scikit-learn.org/stable/

Classifier	Micro			Macro			TTF
	P	R	F1	P	R	F1	(s)
SVM*	0.884	0.426	0.567	0.293	0.155	0.194	0.202
SVM	0.872	0.487	0.616	0.309	0.194	0.228	**0.130**
LR*	0.953	0.120	0.209	0.105	0.028	0.042	1.194
LR	0.932	0.197	0.321	0.137	0.049	0.069	1.159
NB*	**0.971**	0.173	0.292	0.087	0.032	0.043	0.155
NB	0.954	0.181	0.303	0.086	0.033	0.046	0.143
J48*	0.596	**0.641**	0.616	0.359	**0.372**	0.349	0.527
J48	0.609	0.637	**0.622**	**0.363**	0.371	**0.355**	0.502

*Classifier using TF-IDF

Table 1. Results from the evaluation of the classifiers (1st iteration).

tence segmentation approach, we restarted the labeling process. In the end, we labeled a larger sample set of 1,132 reviews with 4,000 sentences. Among them, 1,364 sentences were assigned to one or more factors.

To avoid the bias related to the labeler interpretation, we decided to evaluate the understanding of the definition of each factor with third parties. To do so, we selected five random sentences from each class labeled by the main researcher (including sentences not assigned to any of the factors). Next, we assessed the level of agreement by calculating Cohen's Kappa [Cohen, 2013] with another researcher, an expert in HCI and UX, who was not involved in the data collection. The external researcher received a CodeBook containing the factors and their definitions to support labeling. The results indicated a substantial agreement between the researchers (*Cohen's d* = 0.663) according to the interpretation of Landis and Koch [1977] in Table 2. We discussed the disagreements and identified improvement possibilities in defining some factors.

Table 2. Strength of agreement associated with kappa statistics according to Landis and Koch [1977].

Statistic	Strength of Agreement
< 0.00	Poor
0.00 – 0.20	Slight
0.21 – 0.40	Fair
0.41 – 0.60	Moderate
0.61 – 0.80	Substantial
0.81 – 1.00	Almost perfect

The factor with the highest level of disagreement was "Customer support." It was because its original definition was "*Users being satisfied with the support they received while using apps*." All the sentences assigned to this factor were from users unsatisfied with customer support, leading to zero agreement between the researchers. In this sense, we refined it as follows: "*Users being satisfied or not with the support they received while using apps*." Another improvement was related to the definition of the Usability factor: "*A usability problem is any aspect of a user interface that is expected to cause users problems concerning some salient usability measure (e.g., learnability, performance, error rate, subjective satisfaction) and that can be attributed to a single design aspect*." Although the other researcher assigned positive aspects related to usability, this definition would address only usability problems. In this sense, we refined it as follows: "*Any aspect of the user interface that can facilitate or cause problems to the user concerning some salient usability measure (e.g., learnability, performance, error rate, subjective satisfaction)*." After improving the definitions of the factors and agreeing on them, a third researcher, also an expert in HCI, reviewed the factors' definitions. Next, we restarted the coding process.

We also reviewed the list of stopwords from the NLTK library. We realized that some of the words in this set would be important for the classifier to identify some factors. Words such as "should," "could," "would," and "please" are informative keywords for the *Improvement Request* factor, while words such as "cannot/can't" and even the word "not" followed by "work" (i.e., "not work") may indicate the existence of a *Bug/Crash*. Thus, we removed such words from the stopwords list. Additionally, we analyzed the output from the feature extractors ordered by frequency to investigate whether there are frequent terms that have no relevance to identifying a factor. During this process, we identified words such as "just", "people", "really", "thing" which have no meaning for the classifier. Thus, we removed them, as they might affect the model's training.

Finally, due to the informal and noisy nature of the language used by end users, we performed two additional preprocessing steps as proposed by Palomba et al. [Palomba *et al.*, 2017]: spell correction and contraction expansion. For spell correction, we applied 'symspellpy'[7], a Python port of Symspell, an open-source spell correction algorithm. For contraction expansion, we used simple matching patterns.

In this second iteration, we achieved promising results. J48 was the best classifier, with 59.3% precision and 62.3% recall. Finally, TF-IDF did not perform better but increased the fit time. Thus, we decided not to apply it in the subsequent iterations.

Classifier	Micro			Macro			TTF
	P	R	F1	P	R	F1	(s)
SVM*	0.874	0.643	0.740	0.561	0.403	0.455	0.205
SVM	0.867	0.713	**0.782**	0.558	0.452	0.489	0.167
LR*	0.940	0.282	0.433	0.304	0.108	0.152	0.987
LR	0.926	0.396	0.554	0.362	0.159	0.210	0.911
NB*	0.948	0.265	0.414	0.182	0.075	0.102	0.127
NB	**0.960**	0.282	0.436	0.201	0.082	0.111	**0.118**
J48*	0.682	**0.777**	0.726	**0.593**	**0.623**	**0.588**	0.698
J48	0.681	0.764	0.719	0.565	0.600	0.564	0.613

*Classifier using TF-IDF

Table 3. Results from the evaluation of the classifiers (2nd iteration).

6.2.5 Third Iteration

In this iteration, we increased the number of instances by focusing on factors with few samples to reduce the bias towards the largest factor. To do so, we looked for keywords from already labeled sentences indicating their association with a given factor. For example, sentences from the "Resource Use" factor usually contain words such as "memory", "drain", "lot of", "space", and "bandwidth". We manually

[7] https://pypi.org/project/symspellpy/

searched these terms, analyzed the sentences, and included them in the training set. In the end, we labeled 545 additional sentences, resulting in 1,677 labeled sentences. The link for the entire dataset is available in the "Availability of data and materials" section.

After classifying new instances, we tested other parameters from the classifiers. As our dataset is imbalanced, we enabled the "class_weight" argument and set it to "balanced" in SVM, LR, and J48 classifiers. By doing so, the classifier adjusts the weight of the class inversely proportional to its number of instances. For NB, we tested the ComplementNB (CNB) [Rennie *et al.*, 2003], a particularly suitable classifier for imbalanced datasets, which is our case. We also tested with combinations of n-grams and a classifier called XGBoost (eXtreme Gradient Boosting). It consists of a scalable and sparsity-aware machine learning algorithm used in many machine learning and data mining challenges with good results [Chen and Guestrin, 2016]. For this classifier, we set the learning parameter to "softmax", as it is designed for multiclass classification.

Overall, weighting the classes improved the performance of the classifiers, especially for Logistic Regression (see Table 4). Regarding Naïve-Bayes, although ComplementNB performed better than MultinomialNB, it still performed poorly. XGBoost, in turn, performed similarly to SVM and LR. However, it required much more time to run.

In general, weighted SVM and weighted LR achieved the best results. Considering that it would proportionally require more time to process as the dataset increases and that precision is more important in our context (given that practitioners do not want to spend time reading reviews that are not related to what they are looking for), we decided to employ weighted SVM classifier in UX-MAPPER.

Classifier	Micro			Macro			TTF
	P	R	F1	P	R	F1	(s)
SVM	0.886	0.658	0.755	0.824	0.570	0.649	0.202
SVM*	0.831	0.769	**0.798**	**0.816**	0.696	0.732	0.176
LR	0.964	0.210	0.344	0.335	0.100	0.146	1.055
LR*	0.809	0.789	**0.798**	0.804	0.724	**0.743**	1.646
MNB	**0.971**	0.151	0.260	0.150	0.047	0.070	0.168
CNB	0.694	0.693	0.692	0.580	0.599	0.566	**0.163**
J48	0.666	0.765	0.711	0.751	0.759	0.734	1.026
J48*	0.626	**0.797**	0.700	0.701	0.773	0.713	0.953
XGBoost	0.798	0.682	0.735	0.808	0.681	0.718	561.5

*class_weight = 'balanced'

Table 4. Results from the evaluation of the classifiers (3rd iteration).

6.3 Feature Extraction Component

We employed two state-of-the-art approaches for this component: SAFE [Johann *et al.*, 2017] and RE-BERT [de Araújo and Marcacini, 2021]. They achieved the best results in previous studies reviewing feature extraction approaches [Dąbrowski *et al.*, 2020; de Araújo and Marcacini, 2021].

6.3.1 Simple Approach for Feature Extraction (SAFE)

This approach was designed to extract features from app descriptions and reviews from app stores. It extracts features based on 18 Part-of-Speech (POS) patterns and sentence patterns. Additionally, the approach identifies enumerations and conjunctions to identify lists of features. It also performs a similarity matching to group similar features using cosine similarity. Unfortunately, the authors did not make the approach publicly available. We tried to contact them by e-mail without success. Thus, we reproduced it based on the information available in the paper [Johann *et al.*, 2017] with some adaptations, as they did not provide implementation details.

In our implementation, the approach begins by splitting the review into sentences using SpaCy. Then, it preprocesses the sentences by removing stopwords and applying lemmatization. To group similar features, we used a clustering algorithm called "fast clustering" from Sentence-BERT, a state-of-the-art sentence, text, and image embeddings that use BERT (Bidirectional Encoder Representations from Transformers) to derive semantically meaningful sentence embeddings [Reimers and Gurevych, 2019]. After clustering similar words, we ordered them by frequency, selecting the most frequent as the main feature to be presented in the tool.

6.3.2 RE-BERT

This approach extends BERT, a pre-trained transformer network that presents state-of-the-art results for many NLP tasks, such as question answering, sentence classification, and sentence-pair regression [Reimers and Gurevych, 2019]. They fine-tuned the BERT model to find significant correlations between the sequence of tokens in a review ($x = (x_1, x_2, , x_T)$) and a sequence of tokens that represents the software requirement ($x_a = (x_1^a, x_2^a, , x_S^a)$), where x_a is a subsequence of size *S* (with *S >= 1*) [de Araújo and Marcacini, 2021].

The training set should be in the BIO format to train the model. The 'B', 'I', and 'O' tags indicate that the token is the beginning, is inside, or is outside the software requirement, respectively. To build the training set, we analyzed 3,000 reviews from three educational apps: Google Classroom, Programming Hub, and SoloLearn.

After implementing SAFE and RE-BERT, we evaluated their performance as described in previous studies [Dąbrowski *et al.*, 2020; de Araújo and Marcacini, 2021], where a feature can match the truth set in three levels: 1) *exact match:* when the feature is the same present in the truth set; 2) *partial match 1:* when part of the feature matches the truth set, and there is at most one word that does not match; and 3) *partial match 2:* when part of the feature matches the truth set, and there is at most two words that do not match. Features with three or more words that do not match the truth set were considered false-positive.

We selected a sample of 200 reviews from Google Classroom and extracted their features manually to build our Oracle. Then, we applied the two approaches to extract these features and compared them with the oracle we built. Table 5 presents the results for each approach according to the matching levels.

	Exact			Partial 1			Partial 2		
	P	R	F1	P	R	F1	P	R	F1
SAFE	.532	**.976**	.689	.548	**.977**	.702	.560	**.978**	.713
RE-BERT	**.707**	.917	**.798**	**.726**	.923	**.813**	**.735**	.927	**.819**

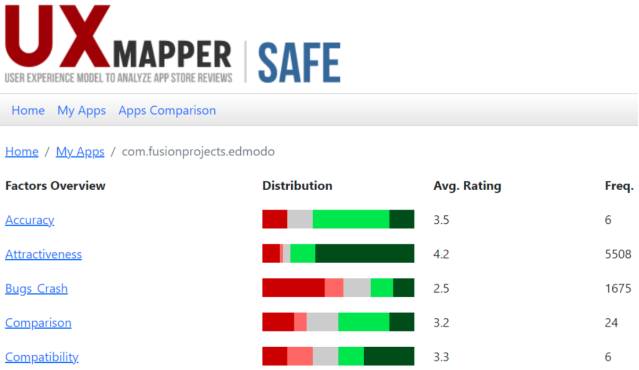
Table 5. Comparison between SAFE and RE-BERT.

RE-BERT achieved the best results, mainly in terms of precision. This is because it extracts features according to what the model learned from the training set. SAFE, in turn, extracts every set of terms that matches the patterns, thus resulting in low precision.

6.4 UX-MAPPER Web Application

For the front-end development, we used Bootstrap[8], one of the most popular front-end open-source toolkits, to develop interface components using HTML, CSS, and JavaScript. To integrate our machine learning model in Python to the Web, we adopted Flask[9] as a back-end engine. It is a lightweight Web framework that provides a set of core libraries for handling common Web development tasks, such as URL routing, template rendering, session management, interactive web-browser debugger, and easy-to-use, flexible application configuration management [Grinberg, 2018]. Finally, to deploy UX-MAPPER, we used Git[10] for version control and Heroku, a Platform as a Service (PaaS) that allows developers to build, run, and operate applications in the cloud.

In the initial screen, practitioners select the app they want to analyze. After selecting the app, UX-MAPPER presents the set of factors analyzed, the distribution of star ratings (in which dark red represents a 1-star rating, and dark green represents a 5-star rating), the average rating of the factor, and the number of reviews associated with it (Figure 4).

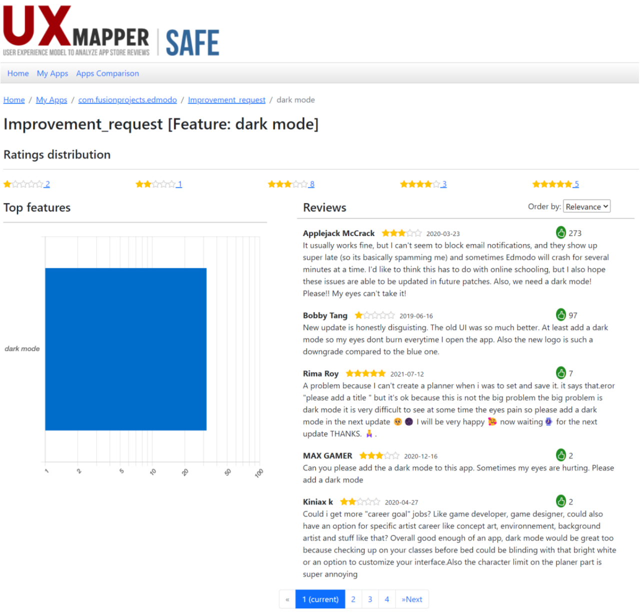


Home / My Apps / com.fusionprojects.edmodo

Factors Overview	Distribution	Avg. Rating	Freq.
Accuracy		3.5	6
Attractiveness		4.2	5508
Bugs_Crash		2.5	1675
Comparison		3.2	24
Compatibility		3.3	6

Figure 4. Part of the factors analyzed by UX-MAPPER.

By clicking on the desired factor, UX-MAPPER shows the top 10 features extracted from the reviews related to this factor and the distribution of the ratings according to the number of stars. The features are ordered by frequency, where the most frequent feature is presented at the top. After selecting the desired feature, UX-MAPPER presents the reviews associated with this feature (Figure 5). The reviews are ordered by relevance by default, i.e., reviews with the greatest number of thumbs up given by other users appear first on the top, similar to the Google Play Store.



Home / My Apps / com.fusionprojects.edmodo / Improvement_request / dark mode

Improvement_request [Feature: dark mode]

Ratings distribution

★☆☆☆☆ 2 ★★☆☆☆ 1 ★★★☆☆ 8 ★★★★☆ 3 ★★★★★ 5

Top features

Reviews

Applejack McCrack ★★★☆☆ 2020-03-23 👍 273
It usually works fine, but I can't seem to block email notifications, and they show up super late (so its basically spamming me) and sometimes Edmodo will crash for several minutes at a time. I'd like to think this has to do with online schooling, but I also hope these issues are able to be updated in future patches. Also, we need a dark mode! Please!! My eyes can't take it!

Bobby Tang ★☆☆☆☆ 2019-06-16 👍 97
New update is honestly disguisting. The old UI was so much better. At least add a dark mode so my eyes dont burn everytime I open the app. Also the new logo is such a downgrade compared to the blue one.

Rima Roy ★★★★★ 2021-07-12 👍 7
A problem because I can't create a planner when i was to set and save it. it says that.eror "please add a title " but it's ok because this is not the big problem the big problem is dark mode it is very difficult to see at some time the eyes pain so please add a dark mode in the next update 🥺 🌑 I will be very happy 🥰 now waiting ⏳ for the next update THANKS. 🙏 .

MAX GAMER ★★★☆☆ 2020-12-16 👍 2
Can you please add the a dark mode to this app. Sometimes my eyes are hurting. Please add a dark mode

Kiniax k ★★☆☆☆ 2020-04-27 👍 2
Could i get more "career goal" jobs? Like game developer, game designer, could also have an option for specific artist career like concept art, environnement, background artist and stuff like that? Overall good enough of an app, dark mode would be great too because checking up on your classes before bed could be blinding with that bright white or an option to customize your interface.Also the character limit on the planer part is super annoying

Figure 5. Reviews associated with the "dark mode" feature.

In this screen, practitioners can also analyze the distribution of the ratings for the selected feature. By doing so, it is possible to identify the impact of this feature. For instance, regarding the "Improvement request" factor, a feature with more reviews with 1 or 2 stars may indicate that it is critical and needs to be prioritized. In turn, a feature in which the reviews are mostly positive (4 or 5 stars) suggests that this feature does not have so much impact on users' experience and should be given lower priority. The practitioner can also switch it to show the most recent reviews first. Finally, practitioners can filter the reviews by the number of stars, which makes it possible to identify the impact of the feature and the reasons behind these ratings.

7 UX-MAPPER Evaluation

In this section, we present the second iteration of the Design Cycle. Due to the pandemic scenario of COVID-19, we conducted all the studies remotely through Google Meet. We conducted two pilot studies before executing our main study.

7.1 First pilot study

Two participants with experience in requirements elicitation participated in this study. They had to analyze Google Classroom reviews and extract requirements that would support the development of a new release focusing on conveying positive UX. The participant analyzed the reviews associated with the feature extracted from the *Attractiveness*, *Bugs/Crash*, and *Improvement request* factors (see Figure 5). We selected these factors as they have the greatest number of reviews. Due to time constraints, each participant explored the first three features and the first five reviews from each factor (45 reviews). Each participant performed the same tasks for each approach (SAFE and RE-BERT) in a cross-over design.

The results revealed that the tasks were very time-consuming. The first participant took 1h45min to extract

[8] https://getbootstrap.com/
[9] https://flask.palletsprojects.com/
[10] https://git-scm.com/

features from the 45 reviews returned by the RE-BERT approach, making the study unfeasible to be conducted with practitioners from the industry, considering they have limited availability. Thus, we decided to reduce the number of reviews from five to three, which resulted in 27 reviews to be analyzed.

The second participant analyzed 27 reviews extracted from the SAFE approach. Even with fewer reviews, the participant took 1h8min to extract features from this approach, which was still high to perform a cross-study design. When analyzing the features extracted, we realized that the results are not directly comparable, as the features extracted by the approaches led to different reviews that resulted in different requirements. Moreover, we did not ask direct questions about the approach. Thus, we could not identify which of them was better from the participant's point of view.

7.2 Second Pilot Study

This study involved four participants who had experience in requirements elicitation. Each participant began interacting with one of the approaches. We asked them to analyze each feature and reflect on whether it would be useful to improve UX. Then, the participant was allowed to click on the feature and visualize its reviews. Next, we presented the features extracted by both approaches, side by side in a PowerPoint presentation, for each of the previous three factors. We asked the participant to analyze each feature and reflect on its meaning to assess whether it is understandable and has the potential to return helpful information to improve the UX of the Google Classroom app. We crossed out the features the participant did not consider understandable or useful during the study (see Figure 6). After assessing all features of all factors, we asked the participant to decide which approach s/he would choose to use. Finally, we asked the participant to answer a post-study questionnaire.

In general, RE-BERT had more features that were considered relevant than SAFE. The results indicate a possible correlation between the number of relevant features and their choices. However, it is also possible that the approach they interacted with had influenced their preference toward that approach. For instance, participants P1 and P3, who interacted with RE-BERT, preferred it, while participant P2, who interacted with SAFE, considered it better. Thus, we had to adjust for the final study to avoid this bias. We also needed more straight-to-the-point metrics, rather than just the number of relevant features, to thoroughly compare the approaches.

7.3 Main Study

We decided to present the features of both approaches side by side before the participant interacts with the tool to reduce the primacy bias. After analyzing the features, the participant interacted with both approaches to explore the features they did not understand or considered irrelevant. We also added three Likert-type questions to directly evaluate the usefulness, easiness, and diversity of the features extracted by each approach and an open question to justify their answers and get qualitative data.

We conducted the study with 14 practitioners who work in the software development industry with experience in requirements engineering and did not participate in the pilot studies. Experience in analyzing user feedback was desirable but not mandatory. Ten participants analyzed user reviews/feedback in at least one project. Among them, three already used automated approaches. The other four participants did not have experience in analyzing user reviews.

In this study, we used the following materials: i) an informed consent form; ii) a characterization questionnaire; iii) a presentation for the participant to assess the features extracted by each approach; iv) the UX-MAPPER tool; and v) a post-study questionnaire with three questions using Likert scale to assess the usefulness, easiness, and diversity of the features extracted by each approach, and an open question for the participants to justify their answers; and vi) the core TAM constructs (Perceived Usefulness, Perceived Ease of Use, and Behavioral Intention) with additional three TAM3 constructs (Job Relevance, Output Quality, and Result Demonstrability) to assess the acceptance of UX-MAPPER [Venkatesh and Bala, 2008]. Given that the perceived usefulness of a system is affected by an individual's judgment on the match between their job goals and the consequences of using the system [Venkatesh and Davis, 2000], we selected these three dimensions to assess participants' perceptions of the relevance of UX-MAPPER in their activities in the industry.

7.3.1 Procedure

First, we introduced the context and motivation of our research. Then, we asked the participant to sign the informed consent form and fill out the characterization questionnaire. Next, we presented the features extracted by both SAFE and RE-BERT approaches for each of the previous three factors side by side. The participant analyzed each feature, and we crossed out those not considered understandable or useful. After assessing the features, the participant had to decide which approach they preferred. Next, we asked the participant to explore the reviews of each feature from both approaches, focusing on the previous features they had crossed out. The participants could explore the tool freely to reflect on their actual usage in a real situation. This exploration investigated whether their opinion on their preferred approach changes by reading the reviews and understanding the features better. Next, we asked the participant whether they would change their opinion on their preferred approach. Finally, we asked the participants to answer the post-study questionnaire.

7.3.2 Results

In this section, we present the results of the main study. First, we present the results of the post-study questionnaire and the participants' assessment of the usefulness of the features extracted by each approach. Next, we present the results on the acceptance of UX-MAPPER by analyzing the answers for each dimension of the TAM3 questionnaire.

Post-study questionnaire: Regarding the amount of information provided by the features extracted, the participants

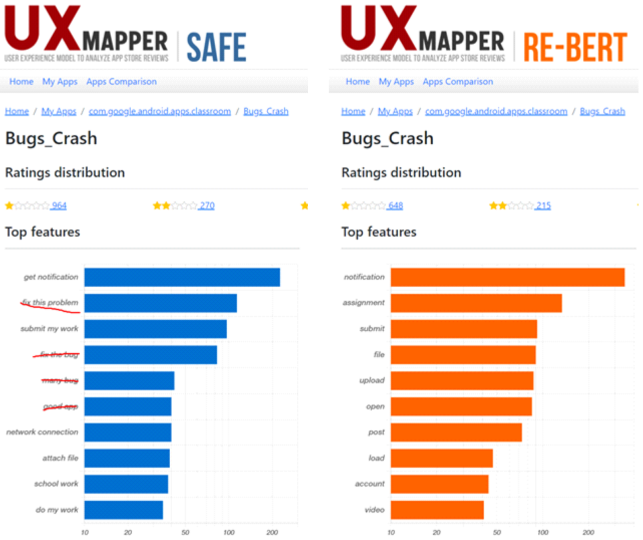


Figure 6. Features considered not understandable or useful by participant P2.

preferred more specific terms for features and functionalities instead of generic ones. Most participants found the features presented by RE-BERT more informative than SAFE to identify opportunities to improve UX, reflected in the number of features the participants considered relevant (Figure 7). In general, participants reported that RE-BERT found more aspects and that they were more specific, concise, and assertive (Figure 8a). In turn, most participants considered generic terms such as "app", for instance, irrelevant, indicating the need for more context when extracting reviews. Regarding SAFE, some participants pointed out that aspects such as "good app", "bad app", "love this app" do not provide useful information (Figure 7a), as they are only general opinions about the app. Other participants also pointed out features that are not clear enough or just some common irrelevant expressions, such as "fix this problem". Since SAFE extracts every set of words that matches the patterns, it is more susceptible to noise, resulting in less relevant features. In turn, some participants considered that these generic terms could be beneficial in identifying issues other than those presented in the graph, which may be useful for exploratory purposes.

Despite the preference for more specific terms, most participants considered "bad experience" and "good experience" extracted by SAFE relevant, although they are also general terms like the ones mentioned before. Participant P4 stated, "*regarding experience, I would check it, as it may have some features we can identify to build the app*". Participant P13 also reported that these aspects "*would show something related to the experience as a whole for the attractiveness of the app*". Some participants considered the aspects SAFE extracted more related to UX, in addition to being more in line with the factors they belong to. Participant P12, for instance, said, "*SAFE has a more emotional language and really shows the user's experience to allow improvements. RE-BERT has some functionalities well applied, but it does not bring users' emotions to improve the app*". These quotations highlight the value of analyzing UX to improve software applications.

In addition to the nature of the reviews, we also identified differences in the amount of information conveyed by the features according to the UX factor under analysis. Regarding the "Bugs/Crash" factor, for instance, RE-BERT performed better, as the features extracted are more specific and mainly related to functionalities (Figure 7b). In turn, SAFE performed the worst. Many participants considered the features generic and redundant. Terms such as "fix the problem", "fix the bug", and "many bug" were considered very similar, not pointing out specific issues that need to be checked. The participants also considered it awkward to have "good app" as a feature from the "Bugs/Crash" factor. After visualizing the reviews, they realized it was because many users praised the app at the beginning of the review and then pointed out the issues they were facing. However, the participants still considered this aspect irrelevant, as the approach did not extract the essential part of the review that comes after the appraisal. Regarding the "Improvement request" factor, SAFE achieved its best results, although still below RE-BERT's performance (Figure 7c). The participants considered it provided more context by using more terms instead of only one as the RE-BERT approach. For example, SAFE returned the "dark mode" feature, while RE-BERT only returned "mode", making it difficult to understand what the latter refers to. Participant P10, for instance, considered all the aspects returned by RE-BERT irrelevant: "*I would not know what 'mode' refers to. 'Button' is also not clear enough. Specifically in this factor, the aspects are not clear what users want. It seems it only threw these words, and I don't know what to do with them*". However, SAFE still presented some generic terms, such as "add a feature", "more feature", "new feature", and "good app", which many participants considered irrelevant.

The second question assessed whether the approaches provide varied and unique features (Figure 8b). The results indicate that the features extracted by SAFE are not diverse and unique compared to RE-BERT. The participants reported that some features from SAFE are redundant (e.g., "fix the problem" and "fix the bug") or variations of the same feature that do not provide much context (e.g., "good app" and "bad app"), which require the practitioner to analyze the comments to interpret them. They also pointed out that its performance on the "Bugs/Crash" and "Attractiveness" factors was poor, given that the aspects they extracted in these factors were too generic (e.g., "many bug", "love this app"). In turn, some participants considered that SAFE performed better in the "Improvement request" factor. Participant P2, for instance, stated "*SAFE was more specific when listing the features for the improvement factor*". Other participants also reported that compliments and improvements are unique, and SAFE captured them well.

The third question assessed the participants' understanding of the extracted features (Figure 8c). In general, the participants did not have difficulty comprehending the features returned by the approaches. Regarding SAFE, some participants reported that the outcomes are clear and easy to understand, while others complained that they seem more like expressions than features, some redundant. Participants also pointed out that they had difficulty understanding the aspect, but after reading the reviews, it made sense. Conversely, some participants considered it a drawback, as they wasted time visualizing the reviews to comprehend the extracted aspect. Regarding RE-BERT, the conciseness of the aspects

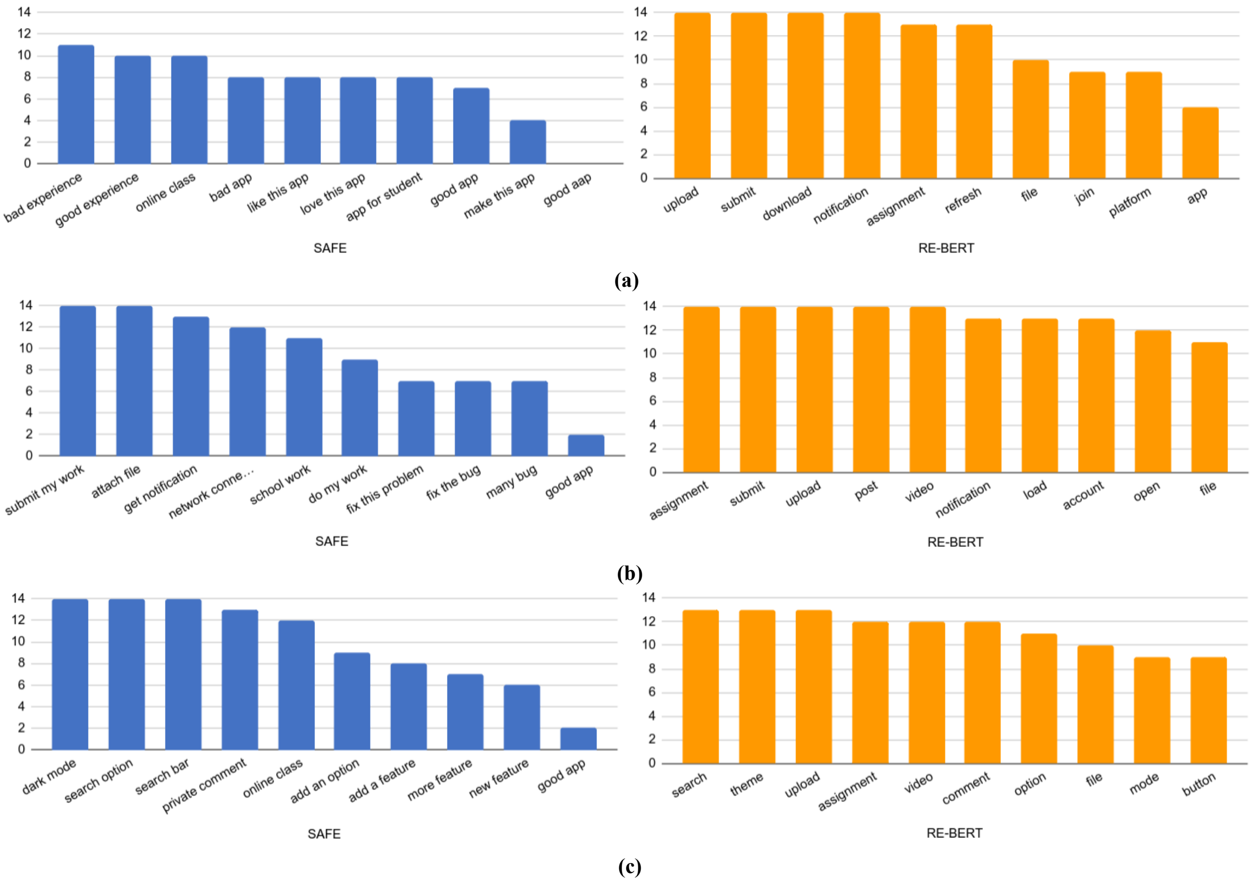


Figure 7. Practitioners perceptions on the usefulness of the extracted features for (a) Attractiveness, (b) Bugs/Crash, and (c) Improvement Request factors.

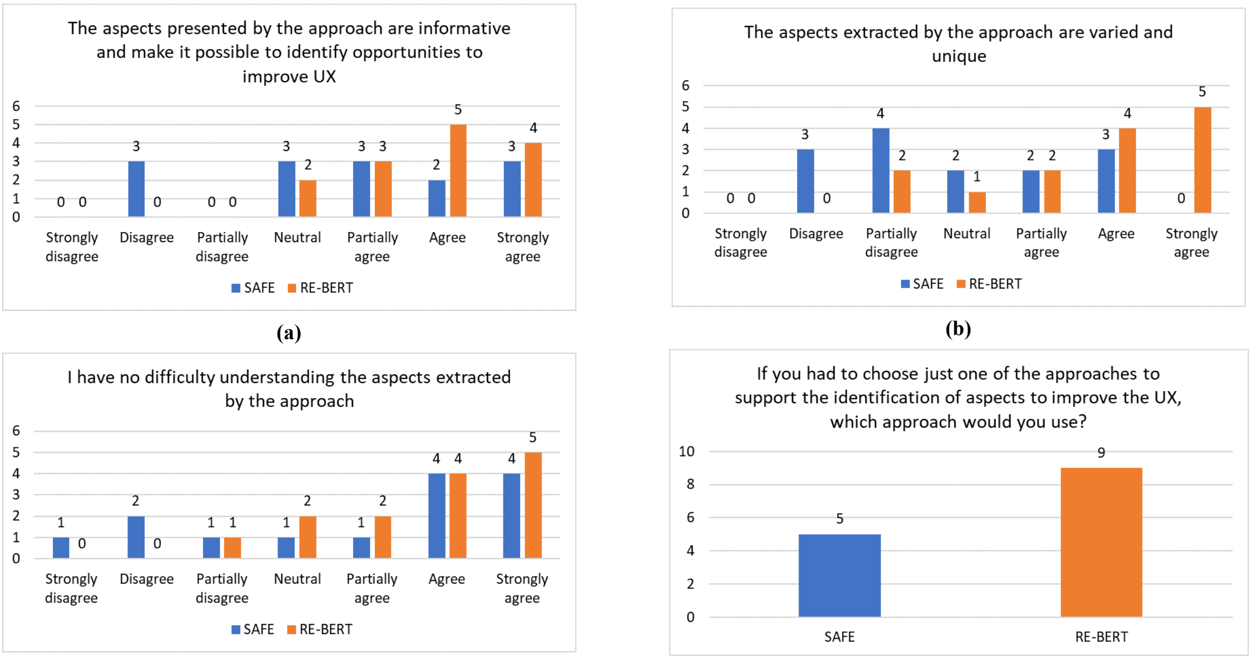
and their focus on more objective aspects made the outcomes easier and clearer to understand. In turn, some participants reported that some features are generic and that sometimes it is not so clear because RE-BERT presents only one word, making it difficult for practitioners without domain knowledge to understand its meaning. Participant P12, for instance, said "*I can understand [the features] due to my experience and knowledge in the area, but people who do not have the minimum [experience and expertise] would have difficulty*". Interestingly, the perception regarding the number of words extracted per feature varied according to the factor. For the *Bugs/Crash* factor, practitioners did not mind having just one term extracted. This indicates that they prefer more straight-to-the-point terms related to functionalities that are not subjective. Identifying which functionalities are causing the bug is essential to making fixes, and generic or subjective terms do not help. By contrast, some participants preferred to have details of what features or functionalities users request in the *Improvement request* factor. Generic terms such as *option* and *button* from RE-BERT were insufficient for them to identify the requested changes.

Regarding the preferred approach (fourth question), nine participants chose RE-BERT and five SAFE (Figure 8d). Most participants preferred RE-BERT because it focuses on functionalities and provides straightforward and non-redundant features. Participants who valued more subjective and emotional aspects tended to choose SAFE. They considered that the features it extracted are more related to UX and more in line with the factors they belong to. In this sense, we identified two stakeholder profiles. One is more concerned with subjective aspects of what users feel, their emotions, and opinions about the app, i.e., the hedonic part of the experience. The other, in turn, focuses more on functionalities and tasks, i.e., the pragmatic part of the experience. Thus, addressing both features is essential to provide a more holistic view of the experience and support practitioners in identifying improvement opportunities.

In summary, practitioners preferred to have an overview of the main points they should look at to improve the app and meet users' needs. Thus, they preferred more specific terms for features and functionalities instead of generic terms requiring further review analysis. Due to this, participants generally found RE-BERT to be more informative, specific, concise, and assertive than SAFE, resulting in fewer features being checked as not useful.

Based on the participants' evaluations of the features' usefulness and the results of the post-study questionnaire, we identified a set of recommendations to provide potentially useful features:

- **The feature extracted should not be redundant:** similarity metrics should be calculated to avoid presenting similar terms;
- **Extract 2-grams or more to provide more context:** single-word features are limited to convey its meaning, thus extracting 2-grams or more would help to understand them before looking at the reviews;
- **Filter general opinions and generic terms:** terms like "good app" and "fix" do not convey much information on what should be improved or fixed and should be re-



(a)

(b)

(c)

(d)

Figure 8. Results of the post-study questionnaire.

moved;

- **Extract features related to functionalities:** most useful features were related to app functionalities, such as "attach file" and "notification". Practitioners preferred straight-to-the-point features that indicate what to improve or fix;
- **Extract UX-related features:** some participants highlighted the importance of considering emotional terms like "bad experience" to understand what affects users' experience. Consider including emotional terms in the analysis to support identifying the most critical features.

TAM3 Questionnaire results: The results from the TAM3 questionnaire indicated a positive acceptance of UX-MAPPER. Given that the TAM3 dimensions comprise different items aimed at capturing a single concept (the dimension itself), we calculated the average score for each evaluated dimension [Sullivan and Artino, 2013], similar to previous works in the HCI field [Alexandrakis *et al.*, 2020; de Sá Siqueira *et al.*, 2024]. Figure 9 presents the distribution of the responses for each questionnaire item. In contrast, Figure 10 presents the average of the ratings for each dimension according to the experience level in analyzing user reviews.

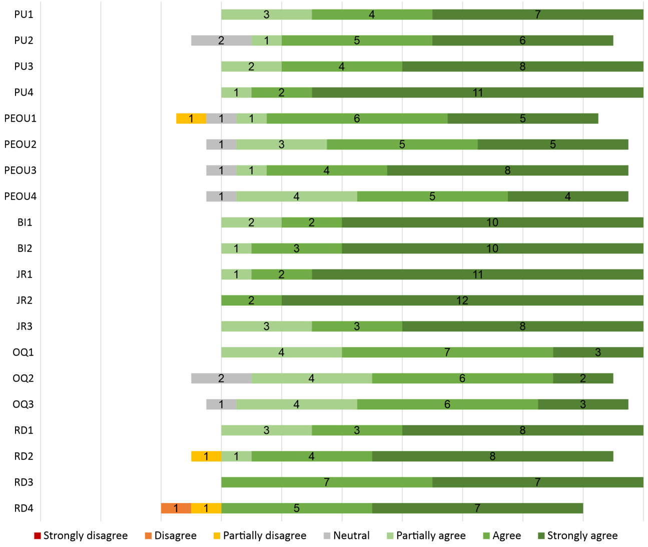


Figure 9. Distribution of the responses for TAM3 questionnaire items.

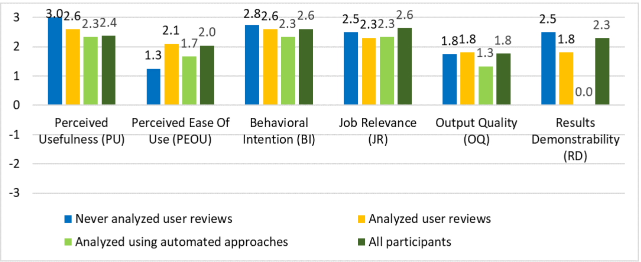


Figure 10. Results from the TAM3 questionnaire by profile.

Regarding Perceived Usefulness (PU), all participants considered it useful in the context of software development (PU4) by improving their performance (PU1) and increasing their effectiveness (PU3). Participants who had never analyzed and had already analyzed user reviews were the most positive regarding the tool. The novelty of an automated approach that reduces the effort to analyze thousands of reviews may have contributed to a more positive evaluation. Participant P13, for instance, stated "*it facilitates organizing and finding the reviews through the factors and features*". Participant P7 also commented, "*I liked it. The classification of the reviews [into factors] is nice. It classifies the reviews into bugs and improvements well*". Participant P5 also commented on the usefulness of feature extraction combined with the star ratings: "*by analyzing the feature together with the star ratings, we can verify its impact on users' satisfaction and, thus, identify which feature to prioritize*". Two participants were neutral when asked whether UX-MAPPER increases their productivity (PU2). Such evaluation reflects their perception regarding the redundancies and generic aspects provided by the approaches, which sometimes requires them to analyze the reviews to comprehend the aspects better.

Perceived Ease Of Use (PEOU) was the second lowest evaluated dimension. Participants without experience analyzing reviews did not find the tool so easy to use. The lack of familiarity with this data type may have affected their understanding of exploring the reviews to extract useful information. Participants who already used automated approaches may have also felt a little confused compared to previous tools with which they were familiar. Some participants, for instance, had difficulty interacting with the tool. In the graph with the aspects extracted, the labels are not clickable, and the tool does not provide clues that the user can click on the bars, such as by changing the cursor to the 'hand' icon. Participant P14, for instance, stated "*the only thing that is counter-intuitive is that usually, the intuition tells us to click on the category name and not in the graphic bar*". Participant P10 also pointed out "*the interaction is not so understandable. The rating distribution graph was strange at first, I couldn't understand it a priori. It could have a caption or change the graphic format [to facilitate its comprehension]*".

Regarding Behavioral Intention (BI), all participants expressed their intention to use UX-MAPPER if it was made available. Participant P6, for instance, commented, "*I hope it becomes available and practitioners begin using it because what matters the most today is listening to users to bring quality, as they are becoming more and more demanding. So, you are providing this information to people by a tool that could allow mitigating any issue that we could identify*". Such a result reflects their perceptions of its usefulness and ease of use, which, according to Davis [1989], predicts actual system usage.

When asked about its relevance to their jobs (JR), all participants unanimously affirmed that UX-MAPPER is relevant. For instance, participant P9, a UX Designer who worked on several projects analyzing user reviews, stated, "*it greatly facilitates the analysis of reviews from app stores. I used to do all these jobs manually. I could not help but think about how the use of the UX-MAPPER could have facilitated the work I did manually*". Participant P14 also stated, "*it would come in handy to analyze tons of reviews for the app developed by our company*". Such a result highlights the potential and benefits of our proposal for practitioners.

UX-MAPPER received the lowest scores for Output Quality (OQ). Such a result reflects the participants' perception regarding the outcomes of the approaches discussed previously. Although they considered the tool useful and relevant for their job, this result highlights room for improving the aspect extraction process. Participant P11, for instance, commented, "*bringing loose words sometimes lose information. Maybe bringing expressions or a short phrase says more than a unique word*". In this sense, the outcomes of the approaches could be improved by providing more context to reduce the need to explore the reviews to understand the aspects better, grouping related terms, and filtering generic expressions.

Results Demonstrability (RD) was the lowest-rated dimension among the participants who had experience using automated text analysis approaches. Such results indicate that the benefits of using UX-MAPPER are not so evident compared to other existing approaches. Participants P3 and P7 considered that they would have difficulty explaining the benefits of UX-MAPPER (RD4). Participant P7 also pointed out that she could not communicate the consequences of using it (RD2). Both participants had already used automated text analysis approaches, which may have served as a baseline for evaluating UX-MAPPER, resulting in lower scores. Regarding participant P3, he identified drawbacks in both feature extraction approaches, which may have affected his perception of demonstrating to others that using UX-MAPPER may be beneficial. Regarding participant P7, she considered that SAFE had much redundancy and generic terms. She was also the only participant who interacted from a smartphone, where the behavior was not tested before the study. The interface became too shrink in portrait mode during her first interactions, which may have affected her perceptions about the demonstrability of the results. Then, we asked her to try using it in landscape mode, which resulted in better visualization, almost similar to the desktop. She was also confused about the meaning of the terms from RE-BERT. For instance, in the "Attractiveness" factor, she had to speculate whether "upload, submit, download" are being talked about positively by users or not, or what users are talking about when referring to the "app" or the "platform": "*it can lead to many possibilities of what it could be*". These issues, added to her previous experience with automated approaches, might have affected her perception of the quality of the output.

In general, the results from the TAM3 questionnaire revealed a positive acceptance of UX-MAPPER. The unanimity on the relevance to their jobs highlights its potential to support the tasks of different roles, from Requirements Engineers and UI/UX Designers to developers and researchers. The participants considered that UX-MAPPER supports identifying the main problems to be fixed and features to be implemented or improved. Classifying the reviews into factors and the features extracted helps organize and find information quickly. It might increase productivity by reducing the effort to extract such information from the reviews manually. In turn, participants who had already used automated approaches considered the benefits of using UX-MAPPER not so apparent. Although they considered it useful and relevant for their jobs, the results indicate room for improvements, mainly in the feature extraction component. There is a need to group similar features, provide more context for the features extracted, and highlight the features in the text to make it easier to identify them. Regarding participants without experience in analyzing user reviews, there is a need to improve the usability of UX-MAPPER to make the interaction more straightforward. The participants pointed out usability issues, such as the lack of clues on whether the graph is clickable, the impossibility of clicking on the feature's name, and the rating distribution graph in the factors overview, which is not intuitive enough.

8 Related Work

Although many studies explore user reviews from app stores, four works are the most similar to our proposal. Hedegaard and Simonsen [Hedegaard and Simonsen, 2014] proposed a tool to extract information from user reviews regarding various UX dimensions from the literature. Bakiu and Guzman [Bakiu and Guzman, 2017] proposed an approach to extract software features from app store reviews and visualize users' satisfaction with these features. In contrast to these works, we did not restrict the classification of the reviews into UX dimensions, but general factors that can affect users' evaluations obtained through a rigorous literature review. Their proposal also has overlapping dimensions and mainly focuses on emotions. This lack of focus on functional aspects may make it difficult for practitioners to identify, for instance, what features users are requesting and their opinions about an update.

Jang and Yi [Jang and Yi, 2017] extracted four UX factors from user reviews of electronic devices and analyzed their impact on satisfaction. The main limitation is that it does not employ Natural Language Processing (NLP) or Machine Learning (ML) techniques to identify and extract the UX aspects but a tool that considers only a single keyword to analyze and identify them. In turn, we employed both NLP and ML techniques to analyze and extract factors from user reviews. Moreover, their focus was different.

McIlroy et al. [McIlroy *et al.*, 2015] proposed classifying negative user reviews into 14 factors. While they performed qualitative analysis on a sample of reviews to identify these factors, we conducted a systematic mapping study of several works to have a broader coverage of influencing factors. In addition to considering all reviews (not only the negative ones), UX-MAPPER presents a set of top features that developers should consider when developing applications.

9 Threats to Validity

Regarding the tool, the imbalanced dataset might have affected the classifier's performance from the factor extraction component. To minimize this threat, we adopted the Iterative Stratification algorithm, which distributes the positive instances of each class among the folds created during the cross-validation process. The positive perception of practitioners using UX-MAPPER may have been influenced by the cultural factor of Brazil, in which people are willing to help each other. We told the participants to be as critical as possible to minimize this bias, given that we wanted to obtain feedback to improve UX-MAPPER. Their previous experience with automated text analysis approaches may have influenced the positive perception. To minimize this bias, we divided the participants into two groups (with and without previous experience with automated approaches) and analyzed the results accordingly.

10 Concluding Remarks

Researchers and practitioners are becoming aware of the importance of User eXperience (UX) in mobile app development. Developing merely usable apps became insufficient to meet users' needs, requiring developers to focus on promoting pleasurable experiences to get a competitive advantage. To do so, it is crucial to understand what factors can lead to positive or negative UX. In this scenario, app store reviews emerged as a valuable source for addressing UX issues by analyzing several self-reports of end-users experiences in the wild. However, analyzing such reviews is costly and time-consuming, highlighting the necessity to develop approaches that automatically analyze such reviews and provide meaningful results.

We conducted this research guided by the question, "*How can we identify the factors affecting users' perceptions of their experience in user reviews from app stores?*". The goal was to support the mobile software development process by developing an approach that helps practitioners identify the factors affecting UX in app store user reviews. To guide the conduction of this research towards the development of an artifact, we applied the Design Science Research (DSR) method. As a result, we proposed UX-MAPPER, an automated approach to analyze app store reviews with a focus on UX.

To evaluate UX-MAPPER, we conducted an empirical study with practitioners from the industry. The goal was to assess our artifact's relevance and usefulness to support them in identifying app features that they should consider during the development process. The results indicated a positive acceptance of UX-MAPPER. The practitioners found it relevant to their jobs and affirmed they would use it if available. They also considered that UX-MAPPER increases their effectiveness and efficiency in software development by providing a set of factors affecting UX and the most frequent features reported by users.

Regarding our research question, we indicate UX-MAPPER to identify the factors affecting UX in app store reviews. In contrast to the work of McIlroy *et al.* [2015], the closest approach to our proposal, we addressed factors extracted from several publications that analyzed various datasets with different apps, allowing us to have a broader coverage of factors affecting UX. We also considered both positive and negative reviews, in addition to extracting features from the reviews to facilitate practitioners identifying the most frequently mentioned issues by users. By using it, practitioners and researchers can analyze the reviews from a given app and investigate what is leading to positive and negative evaluations. The results of the empirical study indicated a positive acceptance of UX-MAPPER, revealing that it is relevant to the practitioners' jobs and supports identifying the main factors affecting the experience.

However, there is still room for improvement regarding the quality of the output of UX-MAPPER. The results revealed, for instance, that some factors require different granularity levels regarding the extracted features. While one-word features (e.g., notification, assignment) extracted by RE-BERT were considered enough to convey the issue in the Bugs/Crash factor, one-word features (e.g., button, mode) were insufficient to indicate improvements in the Improvement Request factor. A combination of RE-BERT's capacity to extract straight-to-the-point terms related to functionalities with the pattern matching of SAFE may help provide more contextualized features, especially for practitioners familiarized with automated tools to analyze reviews. These participants felt the need to refine the features extracted by the tool by removing redundancies and providing more context. An analysis of the approaches used previously by these participants may also bring insights into new functionalities to be included in UX-MAPPER.

Regarding the possibilities envisioned in the Introduction section, our research can contribute to practitioners and researchers as follows:

- **(i) Minimize bias in UX evaluations:** By uncovering the effects of different factors on UX, researchers can plan their studies considering these factors beforehand to minimize their effects. The results of our studies revealed, for instance, factors related to the user characteristics (e.g., gender, culture, previous experience, and particularities of the public of a given type of app, such as health monitoring). Researchers could consider these factors to stratify their samples and analyze the data according to these factors, allowing them to understand the results better and reduce possible bias;
- **(ii) Create UX evaluation techniques that consider these factors:** The factors identified in this research can be a starting point to develop techniques that include these factors in UX evaluations. Questions regarding battery drain, cost, as well as metrics that compare the app with competing ones are not common in UX evaluation techniques and could add value when assessing UX;
- **(iii) Avoid rework by considering the factors beforehand:** By knowing the key factors affecting UX, practitioners can design products that convey a positive UX. The results of our systematic mapping revealed that some factors have different levels of importance according to the type of app or polarity. Usability, for instance, had little effect on the overall rating, but a bad one reduced it significantly. This indicates that usability is not a plus but a critical factor that all apps should meet. In turn, Privacy and Ethical had the greatest negative impact on UX. Thus, developers should avoid collecting or sharing personal data or adopting any mechanism that invades user privacy when developing their apps;
- **(iv) Support the redesign of an app by identifying the impact of the factors affecting UX:** Our research revealed several factors that could affect UX. Identifying the most critical factors reflected on users' ratings through UX-MAPPER could support practitioners in prioritizing those that impact UX the most. Analyzing reviews from competing apps could also provide insights into features to be added and faults to be avoided in new versions. Researchers can also benefit from the results of UX-MAPPER by using it to investigate the effects of these factors in different types of apps. The results could provide valuable information on how these factors affect UX and their importance according to the app type. Researchers could also design weighted UX evaluation methods, such as the one proposed by Lynch *et al.* [2013], who developed a weighted heuristic evaluation to evaluate websites for older adults based on their preferences.

The main contributions of this research are as follows:

- An ad-hoc literature review highlighting publications reporting contradictory results in UX evaluations [Nakamura *et al.*, 2019a], e.g., users evaluate UX positively even when facing many interaction problems and expressing negative emotions. Our findings indicated that other factors may have affected users' perception of the experience, leading to contradictory results.
- Empirical evidence of factors affecting users' perception of the experience and their correlation with positive and negative evaluations [Nakamura *et al.*, 2023].
- A secondary study [Nakamura *et al.*, 2022] addressing publications investigating factors that could affect users' perception of the experience, which implied in: i) an overview of the state-of-the-art on analyzing user reviews from app stores with a focus on UX; ii) a set of factors that could affect users' perception of their experience with mobile applications and their effects; iii) an overview of the methods employed to analyze the reviews; iv) research gaps, challenges, and opportunities for future work with implications to both practitioners and researchers;
- The development of an approach (UX-MAPPER) that automatically analyzes user reviews from app stores to identify the factors affecting UX and extract the main features to support practitioners in identifying improvement opportunities;
- A labeled dataset of user reviews, which can serve as the basis for training ML models and support the development of new tools in the UX field;
- Empirical evidence regarding the usefulness, relevance, and acceptance of using automated approaches to analyze app store reviews from practitioners' perspective [Nakamura *et al.*, 2021];
- A set of recommendations to provide potentially useful features for practitioners;
- A comprehensive analysis of the TAM questionnaire added with three dimensions from the TAM3 questionnaire (Job Relevance, Output Quality, and Result Demonstrability) triangulated with qualitative data, which can serve as the basis for future studies involving the acceptance of a technology with practitioners from the industry;
- A methodology for the development of artifacts based on DSR, which could serve as the basis for the development of new artifacts, especially UX evaluation tools and methods;
- Dissemination of the results and the knowledge obtained during the conduction of this research through publication in journal papers and conference proceedings.

As future improvements in the tool, we aim to *improve the feature extraction component* to extract more relevant features by considering the recommendations presented in Section 7.2.3. We also plan to include new functionalities such as a *temporal analysis of the reviews:* currently, UX-MAPPER does not have an option to define the time slice to be analyzed. By allowing defining intervals, it would be possible to identify tendencies and variations of the factors and features over time. A line graph with the frequencies of reviews would also be useful to visualize peaks that could indicate an event that led to an increase in the number of users doing reviews. Another new functionality to be added is the *comparison of apps and categories:* a comparison between apps would be useful for benchmarking purposes, as well as to identify improvement opportunities, strengths, and weaknesses of the apps analyzed. A comparison between categories would also make it possible to determine which factors and features are common/essential and which are not according to the type of app. Based on these findings, researchers could create guidelines for developing this type of application. Finally, we plan to *improve UX-MAPPER's usability and compatibility with mobile devices:* the results of our study revealed some usability problems that affected participants' interaction. We could apply usability and UX evaluation techniques to assess UX-MAPPER to have a more thorough analysis for improving its interface.

Regarding future research, we identified the following possibilities: i) *Identify the influence of cultural and gender aspects in reviews:* previous works indicate that culture can affect evaluations [Guzman *et al.*, 2018] but not gender [Guzman and Paredes Rojas, 2019]. However, there is a need to investigate it in a broader context. Researchers could also explore whether these factors influence how users write their reviews (e.g., tonality, readability) and how they affect ML models' results; ii) *Investigate the generalizability of UX-MAPPER to other contexts:* further studies could investigate whether reviews written by users in different sources, such as social networks (e.g., Facebook, Twitter), can also serve as input for being analyzed by UX-MAPPER. Researchers could also investigate its adequacy in analyzing reviews from other domains, such as software products in general.

Declarations

Acknowledgements

We thank all the participants in the empirical studies and USES Research Group members for their support.

Funding

This research was partially supported by Amazonas State Research Support Foundation (FAPEAM) through the POSGRAD project 2024-2025, by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES-PROEX) - Finance Code 001, and by the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq - Brazil) through processes 314797/2023-8 and 303443/2023-5. The present work is also the result of the Research and Development (R&D) project 001/2020, signed with the Federal University of Amazonas and FAEPI, Brazil, which has funding from Samsung, using resources from the Informatics Law for the Western Amazon (Federal Law nº 8.387/1991), and its disclosure is in accordance with article 39 of Decree N°. 10.521/2020.

Authors' Contributions

TC and ECCO contributed to the conception and validation of this work. TC, ECCO, and EHTO contributed to writing (Review & Editing) and supervision. WTN performed the experiments and is this manuscript's main contributor and writer. All authors read and approved the final manuscript.

Competing interests

The authors declare that they do not have any competing interests.

Availability of data and materials

All the datasets and raw data used in this study are available in the Supplementary Materials section of the article at https://doi.org/10.5753/jis.2025.4099

References

Al Omran, F. N. A. and Treude, C. (2017). Choosing an nlp library for analyzing software documentation: A systematic literature review and a series of experiments. In *2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR)*. IEEE. DOI: https://doi.org/10.1109/msr.2017.42.

Alexandrakis, D., Chorianopoulos, K., and Tselios, N. (2020). Older adults and web 2.0 storytelling technologies: Probing the technology acceptance model through an age-related perspective. *International Journal of Human–Computer Interaction*, 36(17):1623–1635. DOI: https://doi.org/10.1080/10447318.2020.1768673.

Alves, R., Valente, P., and Nunes, N. J. (2014). The state of user experience evaluation practice. In *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational*, NordiCHI '14. ACM. DOI: https://doi.org/10.1145/2639189.2641208.

Bakiu, E. and Guzman, E. (2017). Which feature is unusable? detecting usability and user experience issues from user reviews. In *2017 IEEE 25th International Requirements Engineering Conference Workshops (REW)*. IEEE. DOI: https://doi.org/10.1109/rew.2017.76.

Bopp, J. A., Mekler, E. D., and Opwis, K. (2016). Negative emotion, positive experience? emotionally moving moments in digital games. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, page 2996–3006, New York, NY, USA. Association for Computing Machinery. DOI: https://doi.org/10.1145/2858036.2858227.

Bradley, M. M. and Lang, P. J. (1994). Measuring emotion: The self-assessment manikin and the semantic differential. *Journal of Behavior Therapy and Experimental Psychiatry*, 25(1):49–59. DOI: https://doi.org/10.1016/0005-7916(94)90063-9.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM*

SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, page 785–794, New York, NY, USA. Association for Computing Machinery. DOI: https://doi.org/10.1145/2939672.2939785.

Cockburn, A., Quinn, P., and Gutwin, C. (2017). The effects of interaction sequencing on user experience and preference. *International Journal of Human-Computer Studies*, 108:89–104. DOI: https://doi.org/10.1016/j.ijhcs.2017.07.005.

Cohen, J. (2013). *Statistical Power Analysis for the Behavioral Sciences*. Routledge. DOI: https://doi.org/10.4324/9780203771587.

Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 13(3):319–340. DOI: https://doi.org/10.2307/249008.

de Andrade Cardieri, G. and Zaina, L. M. (2018). Analyzing user experience in mobile web, native and progressive web applications: A user and hci specialist perspectives. In *Proceedings of the 17th Brazilian Symposium on Human Factors in Computing Systems*, IHC '18, New York, NY, USA. Association for Computing Machinery. DOI: https://doi.org/10.1145/3274192.3274201.

de Araújo, A. F. and Marcacini, R. M. (2021). Rebert: automatic extraction of software requirements from app reviews using bert language model. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, SAC '21, page 1321–1327, New York, NY, USA. Association for Computing Machinery. DOI: https://doi.org/10.1145/3412841.3442006.

de Sá Siqueira, M. A., Müller, B. C. N., and Bosse, T. (2024). When do we accept mistakes from chatbots? the impact of human-like communication on user experience in chatbots that make mistakes. *International Journal of Human–Computer Interaction*, 40(11):2862–2872. DOI: https://doi.org/10.1080/10447318.2023.2175158.

Durelli, V. H. S., Durelli, R. S., Endo, A. T., Cirilo, E., Luiz, W., and Rocha, L. (2018). Please please me: does the presence of test cases influence mobile app users' satisfaction? In *Proceedings of the XXXII Brazilian Symposium on Software Engineering*, SBES '18, page 132–141, New York, NY, USA. Association for Computing Machinery. DOI: https://doi.org/10.1145/3266237.3266272.

Dąbrowski, J., Letier, E., Perini, A., and Susi, A. (2020). Mining user opinions to support requirement engineering: An empirical study. In *Advanced Information Systems Engineering*, page 401–416. Springer International Publishing. DOI: https://doi.org/10.1007/978-3-030-49435-3_25.

Grinberg, M. (2018). *Flask web development: developing web applications with python*. O'Reilly Media, Inc.

Gutwin, C., Rooke, C., Cockburn, A., Mandryk, R. L., and Lafreniere, B. (2016). Peak-end effects on player experience in casual games. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, page 5608–5619, New York, NY, USA. Association for Computing Machinery. DOI: https://doi.org/10.1145/2858036.2858419.

Guzman, E. and Maalej, W. (2014). How do users like this feature? a fine grained sentiment analysis of app reviews. In *2014 IEEE 22nd International Requirements Engineering Conference (RE)*, pages 153–162. DOI: 10.1109/RE.2014.6912257.

Guzman, E., Oliveira, L., Steiner, Y., Wagner, L. C., and Glinz, M. (2018). User feedback in the app store: a cross-cultural study. In *Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Society*, ICSE-SEIS '18, page 13–22, New York, NY, USA. Association for Computing Machinery. DOI: https://doi.org/10.1145/3183428.3183436.

Guzman, E. and Paredes Rojas, A. (2019). Gender and user feedback: An exploratory study. In *2019 IEEE 27th International Requirements Engineering Conference (RE)*, pages 381–385. DOI: https://doi.org/10.1109/RE.2019.00049.

Hassenzahl, M. (2007). The hedonic/pragmatic model of user experience. *Towards a UX manifesto*, 10:2007.

Hedegaard, S. and Simonsen, J. G. (2014). Mining until it hurts: automatic extraction of usability issues from online reviews compared to traditional usability evaluation. In *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational*, NordiCHI '14, page 157–166, New York, NY, USA. Association for Computing Machinery. DOI: https://doi.org/10.1145/2639189.2639211.

Hevner, A. and Chatterjee, S. (2010). *Design Research in Information Systems: Theory and Practice*. Springer US. DOI: https://doi.org/10.1007/978-1-4419-5653-8.

Hevner, A. R. (2007). A three cycle view of design science research. *Scandinavian journal of information systems*, 19(2):4.

Jang, J. and Yi, M. Y. (2017). Modeling user satisfaction from the extraction of user experience elements in online product reviews. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '17, page 1718–1725, New York, NY, USA. Association for Computing Machinery. DOI: https://doi.org/10.1145/3027063.3053097.

Johann, T., Stanik, C., Alizadeh B., A. M., and Maalej, W. (2017). Safe: A simple approach for feature extraction from app descriptions and app reviews. In *2017 IEEE 25th International Requirements Engineering Conference (RE)*, pages 21–30. DOI: https://doi.org/10.1109/RE.2017.71.

Landis, J. R. and Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159. DOI: https://doi.org/10.2307/2529310.

Law, E. L.-C., Van Schaik, P., and Roto, V. (2014). Attitudes towards user experience (ux) measurement. *International Journal of Human-Computer Studies*, 72(6):526–541. DOI: https://doi.org/10.1016/j.ijhcs.2013.09.006.

Lynch, K. R., Schwerha, D. J., and Johanson, G. A. (2013). Development of a weighted heuristic for website evaluation for older adults. *International Journal of Human–Computer Interaction*, 29(6):404–418. DOI: https://doi.org/10.1080/10447318.2012.715277.

Maalej, W., Kurtanović, Z., Nabil, H., and Stanik, C. (2016). On the automatic classification of app reviews. *Requir. Eng.*, 21(3):311–331. DOI: https://doi.org/10.1007/s00766-016-0251-9.

Marques, L., Matsubara, P. G., Nakamura, W. T., Ferreira, B. M., Wiese, I. S., Gadelha, B. F., Zaina, L. M., Redmiles, D., and Conte, T. U. (2021). Understanding ux better: A new technique to go beyond emotion assessment. *Sensors*, 21(21). DOI: https://doi.org/10.3390/s21217183.

McIlroy, S., Ali, N., Khalid, H., and E. Hassan, A. (2015). Analyzing and automatically labelling the types of user issues that are raised in mobile app reviews. *Empirical Software Engineering*, 21(3):1067–1106. DOI: https://doi.org/10.1007/s10664-015-9375-7.

Nakamura, W., Marques, L., Ferreira, B., Barbosa, S., and Conte, T. (2020). To inspect or to test? what approach provides better results when it comes to usability and ux? In *Proceedings of the 22nd International Conference on Enterprise Information Systems*, pages 487–498. SCITEPRESS - Science and Technology Publications. DOI: https://doi.org/10.5220/0009367904870498.

Nakamura, W. T., C. de Oliveira, E. C., H. T. de Oliveira, E., and Conte, T. (2024). Ux-mapper: A user experience method to analyze app store reviews. In *Proceedings of the XXII Brazilian Symposium on Human Factors in Computing Systems*, IHC '23, New York, NY, USA. Association for Computing Machinery. DOI: https://doi.org/10.1145/3638067.3638109.

Nakamura, W. T., de Oliveira, E. C., de Oliveira, E. H., Redmiles, D., and Conte, T. (2022). What factors affect the ux in mobile apps? a systematic mapping study on the analysis of app store reviews. *J. Syst. Softw.*, 193(C). DOI: https://doi.org/10.1016/j.jss.2022.111462.

Nakamura, W. T., de Oliveira, E. H. T., and Conte, T. (2019a). Negative emotions, positive experience: What are we doing wrong when evaluating the ux? In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI EA '19, page 1–6, New York, NY, USA. Association for Computing Machinery. DOI: https://doi.org/10.1145/3290607.3313000.

Nakamura, W. T., de Souza, J. C., Teixeira, L. M., Silva, A., da Silva, R., Gadelha, B., and Conte, T. (2021). Requirements behind reviews: How do software practitioners see app user reviews to think of requirements? In *Proceedings of the XX Brazilian Symposium on Software Quality*, SBQS '21, New York, NY, USA. Association for Computing Machinery. DOI: https://doi.org/10.1145/3493244.3493245.

Nakamura, W. T., Marques, L. C., Redmiles, D., de Oliveira, E. H. T., and Conte, T. (2023). Investigating the influence of different factors on the ux evaluation of a mobile application. *International Journal of Human–Computer Interaction*, 39(20):3948–3968. DOI: https://doi.org/10.1080/10447318.2022.2108658.

Nakamura, W. T., Marques, L. C., Rivero, L., Oliveira, E. H. T. d., and Conte, T. (2019b). Are scale-based techniques enough for learners to convey their ux when using a learning management system? *Revista Brasileira de Informática na Educação*, 27(1):104–131. DOI: https://doi.org/10.5753/rbie.2019.27.01.104.

Nalepa, J. and Kawulok, M. (2018). Selecting training sets for support vector machines: a review. *Artificial Intelligence Review*, 52(2):857–900. DOI: https://doi.org/10.1007/s10462-017-9611-1.

Palomba, F., Salza, P., Ciurumelea, A., Panichella, S., Gall, H., Ferrucci, F., and De Lucia, A. (2017). Recommending and localizing change requests for mobile apps based on user reviews. In *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)*, pages 106–117. DOI: https://doi.org/10.1109/ICSE.2017.18.

Panichella, S., Di Sorbo, A., Guzman, E., Visaggio, C. A., Canfora, G., and Gall, H. C. (2015). How can i improve my app? classifying user reviews for software maintenance and evolution. In *2015 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 281–290. DOI: https://doi.org/10.1109/ICSM.2015.7332474.

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics. DOI: https://doi.org/10.18653/v1/D19-1410.

Rennie, J. D. M., Shih, L., Teevan, J., and Karger, D. R. (2003). Tackling the poor assumptions of naive bayes text classifiers. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, page 616–623. AAAI Press.

Rivero, L. and Conte, T. (2017). A systematic mapping study on research contributions on ux evaluation technologies. In *Proceedings of the XVI Brazilian Symposium on Human Factors in Computing Systems*, IHC '17, New York, NY, USA. Association for Computing Machinery. DOI: https://doi.org/10.1145/3160504.3160512.

Sagnier, C., Loup-Escande, E., and Vallèry, G. (2020). Effects of gender and prior experience in immersive user experience with virtual reality. In Ahram, T. and Falcão, C., editors, *Advances in Usability and User Experience*, pages 305–314, Cham. Springer International Publishing. DOI: https://doi.org/10.1007/978-3-030-19135-1_30.

Santiago, M. T. and Marques, A. B. (2023). Exploring user reviews to identify accessibility problems in applications for autistic users. *Journal on Interactive Systems*, 14(1):317–330. DOI: https://doi.org/10.5753/jis.2023.3238.

Schrepp, M., Hinderks, A., and Thomaschewski, J. (2017). Design and evaluation of a short version of the user experience questionnaire (ueq-s). *International Journal of Interactive Multimedia and Artificial Intelligence*, 4(6):103. DOI: https://doi.org/10.9781/ijimai.2017.09.001.

Sechidis, K., Tsoumakas, G., and Vlahavas, I. (2011). On the stratification of multi-label data. In *Machine Learning and Knowledge Discovery in Databases*, page 145–158. Springer Berlin Heidelberg. DOI: https://doi.org/10.1007/978-3-642-23808-6_10.

Sullivan, G. M. and Artino, A. R. (2013). Analyzing and interpreting data from likert-type scales. *Journal of Graduate Medical Education*, 5(4):541–542. DOI: https://doi.org/10.4300/jgme-5-4-18.

Venkatesh, V. and Bala, H. (2008). Technology acceptance model 3 and a research agenda on interventions. *Decision Sciences*, 39(2):273–315. DOI: https://doi.org/10.1111/j.1540-5915.2008.00192.x.

Venkatesh, V. and Davis, F. D. (2000). A theoretical extension of the technology acceptance model: Four longitudinal field studies. *Management Science*, 46(2):186–204. DOI: https://doi.org/10.1287/mnsc.46.2.186.11926.

Wieringa, R. J. (2014). *Design Science Methodology for Information Systems and Software Engineering*. Springer Berlin Heidelberg. DOI: https://doi.org/10.1007/978-3-662-43839-8.