PODER EXECUTIVO
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

UX-MAPPER: A User Experience Method to Analyze App Store Reviews

WALTER TAKASHI NAKAMURA

Manaus
May, 2022

PODER EXECUTIVO
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

WALTER TAKASHI NAKAMURA

UX-MAPPER:
A User eXperience Method to Analyze App Store Reviews

Doctoral dissertation presented to the Informatics Postgraduate Program (*Programa de Pós-graduação em Informática*) – PPGI, at Universidade Federal do Amazonas (UFAM), as one of the requirements to achieve the PhD in Informatics degree.

Advisor: Ph.D. Tayana Uchôa Conte
Co-advisors: Ph.D. Elaine H. T. de Oliveira
Ph.D. Edson César C. de Oliveira

Manaus
May, 2022

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

N163u

Nakamura, Walter Takashi
UX-MAPPER: A User eXperience Method to Analyze App Store Reviews / Walter Takashi Nakamura . 2022
202 f.: il. color; 31 cm.

Orientadora: Tayana Uchôa Conte
Coorientadora: Elaine Harada Teixeira de Oliveira
Coorientador: Edson César Cunha de Oliveira
Tese (Doutorado em Informática) - Universidade Federal do Amazonas.

1. User experience. 2. User reviews. 3. Influencing factors. 4. Machine learning. 5. App stores. I. Conte, Tayana Uchôa. II. Universidade Federal do Amazonas III. Título

PODER EXECUTIVO
MINISTÉRIO DA EDUCAÇÃO
INSTITUTO DE COMPUTAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

UFAM

FOLHA DE APROVAÇÃO

"UX-MAPPER: A User eXperience Method to Analyze App Store Reviews"

WALTER TAKASHI NAKAMURA

Tese de Doutorado defendida e aprovada pela banca examinadora constituída pelos Professores:

TConte

Profa. Tayana Uchoa Conte - PRESIDENTE

Prof. Bruno Freitas Gadelha - MEMBRO INTERNO

Prof. Igor Fabio Steinmacher - MEMBRO EXTERNO

Igor Scaliante Wiese

Prof. Igor Scaliante Wiese - MEMBRO EXTERNO

Isabela Gasparini

Profa. Isabela Gasparini - MEMBRO EXTERNO

Manaus, 13 de Maio de 2022

To my beloved family,
for all the love and encouragement
to pursue my dreams.

ACKNOWLEDGMENT

First of all, I thank God for the gift of life, without which I could not be doing this work. For the blessings, He has poured into my life, mainly because such special people surround me. He gives me the strength to overcome obstacles, face challenges, and move forward toward my goals.

I thank my dear mother, Cecilia Nakamura, for all her love, zeal, and affection. Thank you for supporting me and encouraging me to pursue my dreams. You are my inspiration and the reason for my existence. My dear father, Hiromu Nakamura, for all the care and love. You are my most outstanding example of wisdom, dedication, resilience, and honesty. All the values I have today are thanks to you. My dear sister, Neiva Nakamura, for always being with me no matter what. Thank you for listening to me when I needed it, for the advice we shared, and for all the adventures we've been through over the years.

I thank my fiancé, Matheus Nogueira, for all his love. You made me want to live intensely again, dream, and plan for the future. Thank you for sharing the good and bad times with me, dreaming and planning the future together, and being by my side whenever I needed it. You make my days happier, lighter. My life is much happier by your side. Thank you for being there for all the special moments of my life. I love you!

I thank my advisor, Tayana Conte, for believing in my potential and walking with me since my master's degree. You were the one that showed me the way of the research and made me fall in love with it ever since. I am grateful for all the hours invested in meetings, orientations, and reviews and all the opportunities you have given me in my academic and professional life. You are my inspiration, and I am very proud to say that you are my advisor. Thank you for listening to me, giving advice when needed, and always motivating me to do my best and never give up. I will never forget all you have done for me over these years. Your passion for teaching, advising, and researching is inspiring. The professor and researcher I am today are all thanks to you.

To my co-supervisor, Elaine de Oliveira, for accompanying me since the master's degree, encouraging me, and hoping for my success. You are so kind and inspiring. Thank you for the great moments we had during the events and for always being there to listen to me, assist, and contribute to my research. To my co-supervisor, Edson César, for his brilliant contributions to this doctoral research, always with a critical view that brings me reflections on what I am researching. Thank you for all your patience and for sharing your knowledge with me.

To the Ph.D. professors, Altigran Soares, and Igor Steinmacher, for their valuable contributions to my doctoral qualification exam. Your considerations and suggestions were essential to directing and improving my research. Thank you very much for taking the time to read and review my work.

To the Ph.D. professors Igor Steinmacher, Igor Wiese, Isabela Gasparini, Bruno Gadelha, Gleison Santos, and José Francisco Neto. I can’t thank you enough for agreeing to participate in my doctoral defense board, even with so little time to read and review my work. I will never forget all your support for my research and academic career. Thank you so much for your valuable considerations and contributions, which have brought me important reflections to improve my research.

To professors Iftekhar Ahmed and David Redmiles for their valuable contributions to my research. Thank you for all the fruitful discussions, sharing your knowledge, and the opportunity to collaborate. I learned a lot!

To all my friends from the USES research group. Thank you for welcoming me from the beginning, for all the knowledge shared, and for all the moments we spent (and still will) together. I am very proud to be part of this beautiful family!

To my sensei Flávio Estevão and my friends Wevertton Santos, Rogério Silva, Ueliton Da Silva, Wiltemberg Lima, Marcos Vieira, and Davi Pereira, for all the support in my academic career and for everything they did for Nakamura Dojô during this period. Thank you also to all my students, alumni, parents and friends for all the support.

Last, but not least, to all the subjects who agreed to participate in the empirical studies. Your participation was essential to making this research happen.

ABSTRACT

User eXperience (UX) is a field that is increasingly attracting the interest of researchers in academia and practitioners in the industry. In a fierce competition scenario, companies seek to develop products that promote unique and satisfying experiences to have a competitive advantage over competitors. Thus, UX evaluations play an important role in developing and evolving interactive software products. However, more than just evaluating the UX, it is equally important to understand what factors affect their perceptions of their experiences. By doing so, practitioners could focus on factors that lead to positive UX while mitigating those that affect UX negatively, which could reduce costs and speed up the development process. However, conducting evaluations to identify these factors is costly as it requires highly trained personnel and many users to perform tasks. In this scenario, reviews from app stores emerged as an alternative to obtain valuable information on factors affecting UX and leading to positive or negative evaluations. This doctoral dissertation proposes an approach called UX-MAPPER (User eXperience Method to Analyze App Store Reviews) to support practitioners in the software development process by analyzing app store reviews to identify the factors leading to positive or negative UX. We followed Design Science Research (DSR), a methodology designed to develop artifacts through three well-defined cycles that ensure the research's novelty, relevance, and rigor. We performed exploratory studies to investigate the problem and assess its relevance, a systematic mapping study to identify the factors that affect users' perceptions about their experience with software applications, and an empirical study to determine the relevance and acceptance of our proposal from practitioners' perspective. The results revealed a positive acceptance of UX-MAPPER. The participants were unanimous in affirming that it is useful for their jobs and that they would use it when it becomes available, highlighting our proposal's usefulness and relevance for the software development and evolution process.

Keywords: user experience, user reviews, influencing factors, machine learning, app stores.

FIGURES INDEX

TABLES INDEX

TABLE OF CONTENTS

CHAPTER 1– INTRODUCTION

This chapter presents an introduction to this research. We contextualize our work in addition to presenting the motivation, research questions, goals, and the methodology we followed. Finally, we present the structure of this qualification text.

1.1. CONTEXT AND MOTIVATION

The success of an interactive product is related to users' willingness to keep using it, which is strongly influenced by the perceived enjoyment of doing so (Cockburn et al., 2017). In the '90s, literature from the experiential market (Pine et al., 1999; Schmitt, 1999) highlighted that a product should no longer be seen simply as a bundle of functional features and benefits but as something that provides experiences (Hassenzahl, 2018b). Researchers realized that customers do not want a merely usable product but that "dazzle their senses, touch their hearts and stimulate their minds" (Schmitt, 1999), shifting the focus to the experiential. Since then, interest has arisen in understanding how people feel due to their engagement with technology (Hassenzahl, 2018a), giving rise to User eXperience (UX) research.

While the concept of usability is more narrow, task-oriented and focused primarily on user cognition and performance, UX, in turn, is more holistic, considering not only pragmatic aspects (task-oriented) but also subjective aspects, such as affect, sensations, emotions and value of user's interaction in everyday life (Law et al., 2009). In a fierce competition scenario, companies have been focusing on design and experience, shifting from technology-driven to convenience, expectations, and needs to develop successful products and services (Alves et al., 2014). Understanding how technology can be used to promote unique, satisfying, and enlightening experiences seems to provide a competitive advantage for business and industry (Alves et al., 2014), leading practitioners and researchers to start debating on how to design products capable of providing positive UX (Ardito et al., 2014).

This concern is even greater for developers of mobile apps. The high demand for mobile devices and the ease of developing such applications led to an exponential growth of the app market in the last decade, increasing the competition to earn a spot on mobile users' devices (Nayebi et al., 2018). Due to the wide variety of apps to choose from, mobile users have developed a low tolerance for faulty and low-quality ones, removing such apps from their devices and replacing them easily (Durelli et al., 2018). In this sense, understanding what factors can affect users' perception of more positive or negative experiences became essential to stay competitive in the market. As a factor, we defined every aspect related to the application or the user associated with a positive, negative, or neutral perception of the experience.

Results from recent studies have indicated the existence of factors that weigh more in the experience and can affect the results of UX evaluations. Users, for example, have been evaluating their UX as positive even when facing many interaction problems and expressing negative emotions (Nakamura et al., 2019). A study by de Andrade Cardieri and Zaina (2018) revealed that users that expressed many negative emotions during their interaction still evaluated the UX as positive when evaluating retrospectively through a questionnaire. In another work, Bopp et al. (2016) found that sadness was the most frequently mentioned emotion in digital games. Conversely, players found such experiences rewarding, leading them to give high ratings when evaluating their appreciation and enjoyment. A study also identified that users might make evaluations based on their perceptions instead of reality. In a work investigating the effect of cross-modal perceptions in an audio-visual interface, Metatla et al. (2016) revealed that users found the audio-visual condition easier than the visual-only condition, although the data showed no improvement in their score.

These issues have several implications for users, practitioners, and researchers. Identifying these influencing factors would allow practitioners to focus on factors that influence UX positively while mitigating the effects of factors that influence UX negatively. Thus, users would benefit from developing products that meet their needs and convey a more positive experience. For practitioners, the lack of information on what factors lead to more positive or negative experiences may result in unnecessary effort to develop features or fix issues that will have a small effect on the UX the app conveys. As UX evaluations may lead practitioners to make different design decisions (Borsci et al., 2015), it is important to identify what factors influence users' subjective assessment of their experience to interpret the results better and plan future releases appropriately. Determining the effect of different factors on UX may support practitioners in defining which ones to prioritize during the development or improvement of their apps. For researchers, the lack of information regarding these factors may lead them to carry out studies without considering their effect, resulting in biased results. For example, all UX dimensions and items that compose existing UX evaluation methods currently have the same weight on the final score. The importance of each dimension might vary according to the type of software product and other variables, such as gender and culture. By identifying how different factors affect users' evaluation, it will be possible to develop approaches that evaluate the UX more accurately and enable the creation of guidelines for the evaluation and development of software that focus on aspects that weigh the most in the experience.

One way of identifying such factors is through UX evaluations. A variety of UX evaluation methods has been proposed in the last decade (Rivero and Conte, 2017; Vermeeren et al., 2010). However, conducting evaluations is costly and time-consuming. It requires highly trained personnel and many users to perform tasks, which may not be feasible, especially in an agile context. In this scenario, user reviews can be a valuable source of information that practitioners can use to extract valuable information to drive the development effort and improve forthcoming releases such as requirements, improvement request, bugs, and experience reports (Guzman and Maalej, 2014). In contrast to the feedback collected from controlled experiments, app stores provide reviews written spontaneously by users worldwide for a variety of apps describing what they liked or hated the most. By analyzing these reviews, practitioners could identify which features to prioritize and what factors are more likely to lead to positive or negative reviews. For instance, Pagano and Maalej (2013) identified that reviews requesting new content are the least critical (4.25 stars on average). This paper's finding indicates that users do not penalize the app so much due to the lack of content, allowing developers to focus on other more critical factors when improving the app. Researchers could use this finding to create UX evaluation techniques that attribute weights to the evaluated items according to their impact on UX to obtain more precise indicators.

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 aims to answer the following question: "*How can we identify the factors affecting users' perceptions of their experience in user reviews from app stores?*".

1.2. RESEARCH GOALS

Our main goal is to support the mobile software development process by automatically identifying the factors that affect UX by analyzing user reviews from app stores.

1.2.1. Specific goals

The specific goals of this research are:

- Provide a body of knowledge regarding different factors that can affect UX in mobile apps;
- Define automated strategies to support the software development process by identifying the factors that lead to more positive or negative reviews;
- Support practitioners in identifying users' most frequently reported app features that they should consider during mobile software development.

1.3. METHODOLOGY

In order to achieve the goals of this research, we applied Design Science Research (DSR). Design Science Research is a research paradigm that consists of an iterative research process that aims at the design and investigation of innovative artifacts, i.e., something created for a practical purpose (Wieringa, 2014), contributing with new knowledge to the body of scientific evidence (Hevner and Chatterjee, 2010). In DSR, the artifact is improved iteratively according to the needs of stakeholders to solve a problem and comprises three cycles: relevance, design, and rigor (Hevner and Chatterjee, 2010). We present the concept behind each cycle and an overview of the steps we performed during each cycle below.

Research opportunities and problems in a given application environment are identified in the **relevance cycle**. The environment in Figure 1.1 refers to where the phenomenon of interest (i.e., the problem) is observed and where the artifact operates (Dresch et al., 2015). 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).

Our previous studies (Nakamura et al., 2019, 2020) motivated this research. We realized that many users still evaluated their UX as positive, even when facing problems that impaired them in performing some tasks. From this previous experience, we performed an initial ad-hoc literature review (Nakamura et al., 2019) to search for other studies that reported similar findings and identify research gaps. We hypothesize that there should be factors that weigh more in the users' perception of the experience, leading to contradictory results. We identified various publications that reported contrasting results from empirical UX evaluations (Bopp et al., 2016; Bruun and Ahm, 2015; de Andrade Cardieri and Zaina, 2018), and studies that aimed to investigate the effect of different factors on users' perception of their experience (Cockburn et al., 2017; Gutwin et al., 2016; Kujala et al., 2017), indicating the interest of the community on the topic.

We began investigating the effect of factors on UX by carrying out an empirical study (CHAPTER 3). This study aimed to investigate the influence of different factors on UX by evaluating a mobile shopping application that uses a chatbot. The findings supported our initial hypothesis, indicating that there are factors that can affect how users perceive their experience and, thus, affect the results. 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 papers that analyzed user reviews from app stores and reported the influence of factors on UX (CHAPTER 4). Our focus on app store reviews is because they are considered the "voice of users" (Guzman and Maalej, 2014), from which practitioners could extract valuable information to improve their app or develop a new one based on the analysis of competing apps. Through a systematic mapping study on the topic, we can summarize the factors that can affect UX and identify which methods have been applied to analyze the effect of these factors on UX. The broad view of a systematic mapping study also 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 reviews studies.

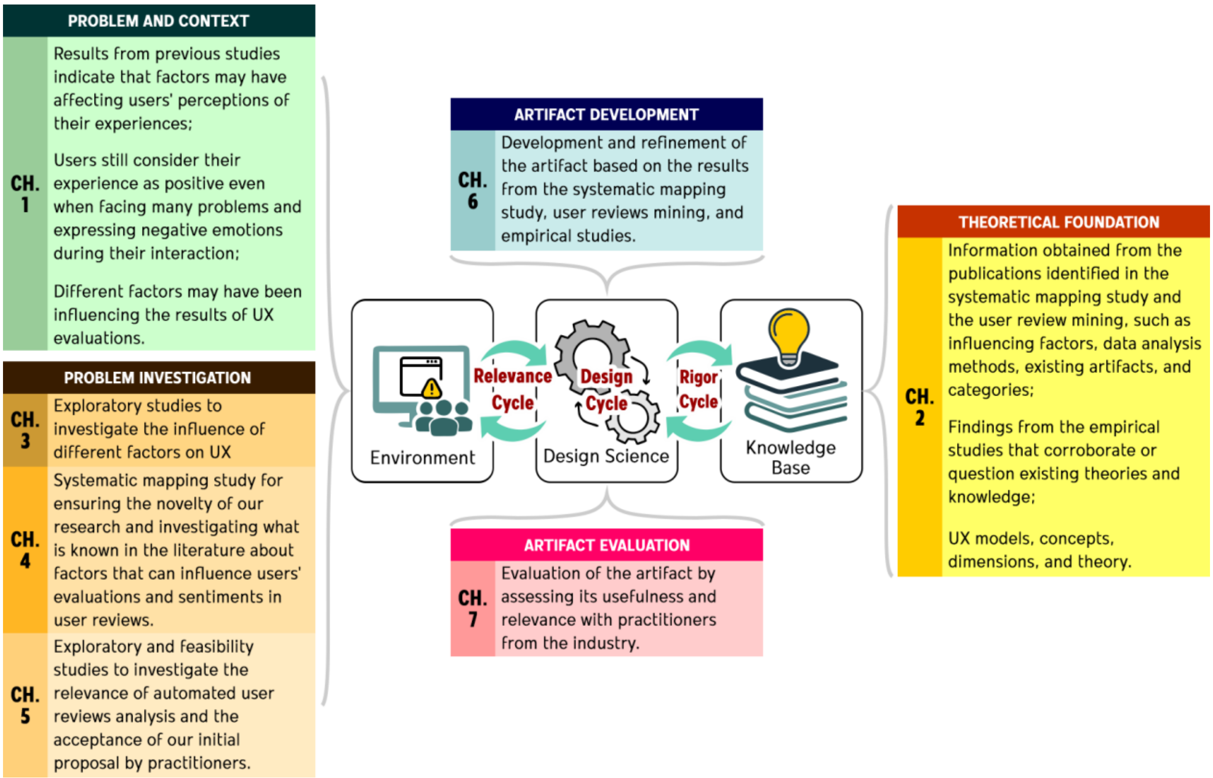

PROBLEM AND CONTEXT
CH. 1
Results from previous studies indicate that factors may have affecting users' perceptions of their experiences;
Users still consider their experience as positive even when facing many problems and expressing negative emotions during their interaction;
Different factors may have been influencing the results of UX evaluations.
PROBLEM INVESTIGATION
CH. 3
Exploratory studies to investigate the influence of different factors on UX
CH. 4
Systematic mapping study for ensuring the novelty of our research and investigating what is known in the literature about factors that can influence users' evaluations and sentiments in user reviews.
CH. 5
Exploratory and feasibility studies to investigate the relevance of automated user reviews analysis and the acceptance of our initial proposal by practitioners.
ARTIFACT DEVELOPMENT
CH. 6
Development and refinement of the artifact based on the results from the systematic mapping study, user reviews mining, and empirical studies.
Environment
Relevance Cycle
Design Cycle
Design Science
Rigor Cycle
Knowledge Base
ARTIFACT EVALUATION
CH. 7
Evaluation of the artifact by assessing its usefulness and relevance with practitioners from the industry.
THEORETICAL FOUNDATION
CH. 2
Information obtained from the publications identified in 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 1.1 - Overview of the DSR cycles employed in this research.

After identifying the factors affecting UX, we conducted an exploratory study to investigate the relevance of automating user reviews analysis from the practitioners' points of view (CHAPTER 5). We performed interviews with practitioners with experience in analyzing user reviews to identify their main activities, the need to analyze user reviews, and the main challenges involved in this process. Based on the findings from this study, we developed an initial proposal and evaluated its acceptance through a feasibility study with practitioners from the industry. The results of this feasibility study allowed us to identify the relevance of our proposal and the main features that should be implemented in our artifact called UX-MAPPER (User eXperience Method to Analyze App Store Reviews).

The **rigor cycle** consists of identifying state of the art to develop an artifact with a solid theoretical foundation. In this cycle, the existing artifacts and processes are identified, as well as the experiences and expertise that define the state of the art in the research application domain, guaranteeing the innovation of the research project (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 from performing the research 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 (CHAPTER 2). Our first source is related to the theory of UX, which involves models, concepts, measures, and dimensions defined by previous works in the literature (Hassenzahl, 2007; Hassenzahl and Tractinsky, 2006; Law et al., 2014; Law and van Schaik, 2010). The second source is the findings from our systematic mapping of studies that analyzed user reviews from app stores focusing on user experience. Finally, we have the experience and results that we obtained by conducting empirical studies 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 through iterative construction and evaluation activities (Hevner and Chatterjee, 2010). In this stage, the artifact is developed based on the theoretical foundation, knowledge, and previous experiences obtained in the rigor cycle (Hevner, 2007). Then, the artifact is evaluated through its application in the environment. The results obtained during the artifact evaluation allow to identify improvement opportunities that will serve as input to the next cycle until a satisfactory design is achieved (Hevner and Chatterjee, 2010).

We developed and refined our artifact iteratively grounded on a solid theoretical foundation obtained from the literature, empirical studies, and previous experiences (CHAPTER 6). To do so, we evaluated different Machine Learning approaches from the literature for classifying the reviews into the factors identified in the systematic mapping study. We also used widely known technologies to support UX-MAPPER. One of them is SpaCy[1], a state-of-the-art natural language processing tool (Al Omran and Treude, 2017), Sentence-BERT[2], 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), and Flask, a lightweight Web framework that provides a set of core libraries for handling common Web development tasks. After developing the tool, we validated it by conducting a study with practitioners from the industry to investigate the relevance and usefulness of UX-MAPPER in the software development context (CHAPTER 7). The results revealed a positive acceptance of UX-MAPPER, its potential to support practitioners on their tasks, and the relevance to software engineering practices.

1.4. ORGANIZATION

This chapter presented an introduction and contextualization of our research, as well as the goals and the methodology we have employed. The remainder of this doctoral dissertation is organized as follows:

Chapter 2 – User Experience and Influencing Factors: this chapter contains the theoretical background of our research. We present the context and definition of UX, an overview of existing evaluation methods and approaches, research opportunities and related work.

Chapter 3 – Investigating Influencing Factors: this chapter presents our first empirical study to investigate the influence of different factors in UX evaluations.

Chapter 4 – Systematic Mapping of Studies Analyzing User Reviews from App Stores: in this chapter, we present the results of our systematic mapping study to investigate state of the art on studies that analyzed user reviews from app stores and presented influencing factors.

[1] https://spacy.io/

[2] https://www.sbert.net/

Chapter 5 – Investigating Practitioners' Perceptions Towards an Automated Approach to Analyze App Store Reviews: in this chapter, we present the results of an exploratory study and a feasibility study to investigate practitioners' acceptance of our proposal.

Chapter 6 – UX-MAPPER Development Process: this chapter presents the iterative process we followed to develop and refine UX-MAPPER, its architecture, and how it works.

Chapter 7 – Evaluating UX-MAPPER from Practitioners' Perspective: in this chapter, we present the conduction of an empirical study with practitioners from the industry to evaluate the acceptance of UX-MAPPER.

Chapter 8 – Conclusions and Future Work: this chapter presents the conclusions derived from the results of this research, the main contributions, and perspectives for future work.

CHAPTER 2 – USER EXPERIENCE AND INFLUENCING FACTORS

In this chapter, we detail the concept of usability and UX. We also describe the role of UX evaluations and the research gaps we identified, which served as a starting point for our proposal.

2.1. USABILITY AND USER EXPERIENCE

According to ISO 9241-11 (2018), usability is defined as "the extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use." In the '80s, during the first wave of Human-Computer Interaction (HCI), the focus of the community was on investigating human capabilities in computer use by employing rigid guidelines, formal methods, and systematic testing (Bødker, 2015; Roto and Lund, 2013) to make technology more usable (Fuchsberger et al., 2012). At that time, users were seen as passive and unmotivated individuals trying to efficiently use the computer (Roto and Lund, 2013). Thus, for many years, effective and efficient goal achievement was the prime objective of Human-Computer Interaction (Hassenzahl, 2018a), making usability one of the main concerns when designing a software product.

Years later, the HCI community entered the era of cognitive science, with an increased emphasis on theory and understanding of what happens in the human mind in terms of information processing (Duarte and Baranauskas, 2016). This second wave brought the idea that the user is an active individual that controls the system, shifting to ease of use and user-friendliness (Roto and Lund, 2013). The focus now was on groups working with a collection of applications (Bødker, 2006), situating the user as an actor who possesses a set of skills and shared practices based on experience (Duarte and Baranauskas, 2016).

Now, in the third wave, the focus is not only on fluent human-computer interaction and getting tasks done but on the role of technology in people's lives and the emotions and experiences that it conveys (Roto and Lund, 2013). Product development was no longer seen as only implementing features and testing their usability but as designing enjoyable products, providing experiences, and supporting fundamental human needs and values (Väänänen-Vainio-Mattila et al., 2008). In this sense, the traditional usability framework, mainly focused on user cognition and performance (Law et al., 2009), became too narrow to represent a holistic vision of human-computer interactions (Lallemand et al., 2015), raising the need for a broader concept that considers non-utilitarian concepts, such as fun, pleasure, hedonic, and ludic (Hassenzahl, 2018a).

The term User eXperience (UX) emerged as an umbrella phrase for new ways of understanding and studying the quality-in-use of interactive products (Bargas-Avila and Hornbæk, 2011). It was originally popularized by Don Norman (1995) in the 90s, becoming widely and quickly disseminated and accepted by the HCI community (Law et al., 2009). However, there is still no consensus about its concept (Lallemand et al., 2015), resulting in several different definitions (Hassenzahl and Tractinsky, 2006; Lallemand et al., 2015; Law et al., 2009). Despite this lack of consensus, researchers and practitioners agree that UX is dynamic, subjective, and context-dependent (Law and van Schaik, 2010).

Different structural models have been proposed to better understand, predict, and reason about UX's processes and establish cause-effect relations. In one of the most influential models proposed by Hassenzahl (2018b), UX is characterized according to two distinguishing dimensions: Pragmatic and Hedonic. Pragmatic Quality is related to a product's perceived ability to support the effective and efficient achievement of tasks, while Hedonic Quality refers to a product's perceived ability to create pleasure through the use [2]. Thus, UX takes a more holistic perspective (Hassenzahl, 2018a), considering not only task-oriented aspects but also exploring subjective aspects that characterizes the experience between human and technology (Lallemand et al., 2015), such as affect, sensations, emotions, and value of user's interaction in everyday life (Law et al., 2009).

In this research, we adopted the definition from Hassenzahl and Tractinsky (2006) to guide the identification of UX-related factors, given that it is, according to Lallemand et al. (2015), the most preferred definition among practitioners and researchers. According to it, UX is "a consequence of a user's internal state (predispositions, expectations, needs, motivation, mood, etc.) the characteristics of the designed system (e.g., complexity, purpose, usability, functionality, etc.) and the context (or the environment) within which the interaction occurs (e.g., organizational/social setting, the meaningfulness of the activity, the voluntariness of use, etc.)."

2.2. UX EVALUATION

Researchers and practitioners from academia and the industry are becoming aware of the importance of providing a good user experience when developing interactive software products (Kou and Gray, 2019). A survey from Lallemand et al. (2015) with 758 participants from 35 nationalities and different fields revealed that 83.9% consider UX as central or very central for their professional work. The results also showed that the primary motivation for the interest in UX is to design better products (51.9%) and to make people happier (21.3%). In this context, evaluating UX is becoming an essential part of developing and evolving interactive software products. Through evaluations, developers can design software more focused on real users, capturing their interaction with the product and reducing its level of rejection (Moreno et al., 2013). However, more than just evaluating the UX, it is equally important to understand what factors affect their perceptions of their experiences. By doing so, practitioners could focus on factors that weigh the most in the experience while mitigating those that affect UX negatively, allowing the reduction of costs and speeding up the development process. In this sense, researchers have been investigating the influence of different factors on UX.

2.2.1. Influencing Factors

Some studies have indicated that the results can be influenced by temporal aspects of the UX evaluation (Kujala and Miron-Shatz, 2015; Soleimani and Law, 2017). In general, users tended to super estimate their experiences when evaluating retrospectively. This phenomenon is known as the *memory-experience gap*, which is defined as the "discrepancy between the average of experienced emotions and the overall evaluation of the experience, which is usually more intense [i.e., stronger] than averaged emotions" (Miron-Shatz et al., 2009). Although it is argued that this phenomenon tends to make negative sentiments more prominent in retrospective evaluations (Miron-Shatz et al., 2009), studies have been pointing out that users might evaluate positively the UX conveyed by a product even when it has many usability problems. Indeed, Bruun and Ahm (2015) revealed that the memory-experience gap is more prominent when users interact with an application with usability problems, giving higher ratings than users who interacted with a usability-free application.

Other studies (Cockburn et al., 2015, 2017; Gutwin et al., 2016) provided evidence that users' preferences and perceptions are affected by the polarity of the most intense event added to the final interaction moment, a phenomenon known as the *peak-end effect*. Cockburn et al. (2017), for instance, aimed to study the influence of the peak-end effect on users' subjective experience. To do so, the participants had to move a set of sliders until matching a given value provided by the application for each slider. Five different interfaces were shown and the number of sliders to be moved varied according to three conditions: i) **end:** focus on the last two interfaces of the series, providing either an increasing (*-end*) or a decreasing (*+end*) workload;

ii) **peak:** there is a sudden increase (*-peak*) or decrease (*+peak*) in the number of sliders in the third interface, but ends with the same amount of sliders; iii) **peak-and-end:** the second and the last interface are manipulated, one with very few sliders (*+peak-and-end*) and other with many sliders in these two interfaces (*-peak-and-end*). The results indicated only a significant change in users' preferences in the *peak-and-end condition*, where a significant majority of the participants preferred the *+peak-and-end condition* over the *-peak-and-end condition*. In another study, Gutwin et al. (2016) evaluated the influence of the peak-end rule on the experience with three casual games. To do so, they conducted two experiments by manipulating the level of difficulty and balance (i.e., the trade-off between challenge and player's skills) along players' interaction sequences to investigate whether they affect user experience regarding four questions: fun, interest, perceived challenge, and willingness to repeat. The study revealed mixed results according to the game. A strong peak-end effect was identified only in the matching game for all four questions. For the other two games, the manipulation only affected the results for the perceived challenge.

Previous experience and the number of problems also seem to affect UX evaluations. Kim et al. (2015) reported that previous experience influences UX ratings. They carried out a longitudinal study by collecting data for a month to evaluate how UX changes over time in a Social Network Service (SNS) app for iPhone. Among the findings, the authors identified that participants who had previous experience with other types of SNSs tended to give significantly higher rating scores for overall UX over time (except for usability and user value, i.e., user's subjective values attached to a product/service) in comparison to those without experience.

In another work, Bolchini et al. (2009) investigated whether usability influences the value-oriented approach of user experience. To do so, they inspected four different websites: two of them characterized by a low level of usability were improved after the inspection, and the other two with good usability worsened. They also inspected the improved and worsened websites. A total of 120 users who had never used these websites, divided into eight groups, participated in the study. First, the researchers asked users to express their level of agreement with brand value statements associated with the brand on a Likert scale. After, each group interacted with one of the versions of the websites by following three scenarios. Finally, they assessed the brand value by using the same initial questionnaire. The results indicated that participants who used a Website with more usability problems tended to express a consistently less positive perception of the brand values than those who used a version with fewer usability problems.

2.2.2. UX Evaluation Methods

There are various UX evaluation methods, each designed for different types of studies (e.g., field study, survey, expert evaluation) and periods of experience (e.g., an episode, long-term evaluation). In contrast to usability evaluation methods that are categorized into inspection, testing, and inquiry (Fernandez et al., 2011), there is no standardized classification of UX evaluation methods, leading to several different categorizations (Bargas-Avila and Hornbæk, 2011; Darin et al., 2019; Pettersson et al., 2018; Rivero and Conte, 2017; Roto et al., 2011; Vermeeren et al., 2010). For simplicity and clarity, we adopted the categories defined by Roto et al. (2011), who classified them into three major categories: observation, psychophysiological measurements, and self-reporting.

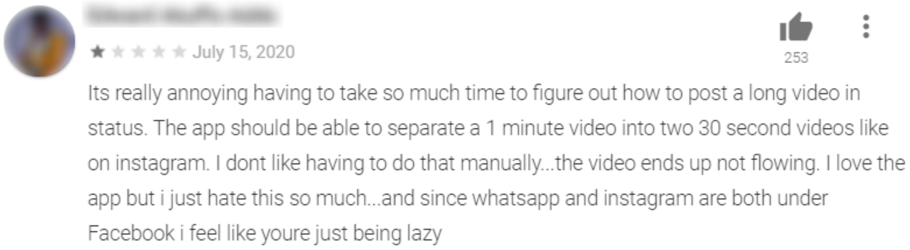
In **observation**, the user interacts with a product while being observed by a moderator in a controlled environment. During the evaluation session, the moderator takes notes of the user's facial, body, and vocal expressions to capture the user's emotions and reactions. An example of an observation method is the Think-Aloud protocol. In this method, the user verbalizes their thoughts and actions while interacting with a product and being observed by a moderator (Alhadreti and Mayhew, 2018). An advantage of this type of method is that it is cheap, and the researcher can obtain specific and instantaneous data according to what they are interested in. However, the user may feel uncomfortable when being observed, which might affect the evaluation results. Another approach is the analysis of facial expressions through the use of emotion heuristics. De Lera and Garreta-Domingo (2007), for instance, proposed ten heuristics to assess users' affective dimension. Researchers can record users' facial expressions using a camera and analyze them afterwards applying the heuristics, which may be useful to complement objective measures collected through usability testing.

Psychophysiological measurements consist of using sensors to get objective measures such as pupil dilatation, heartbeat, and skin conductivity, which can be used, for instance, to detect changes in user's emotions or behavior (Lallemand and Koenig, 2017). An advantage of this type of method is that it can detect even small variations in users' emotions during the interaction. However, connecting devices such as electroencephalogram and eye-tracking to their bodies may make users feel uncomfortable and change their behavior, which might affect the results.

Finally, **self-reporting** consists of the user evaluating their UX using methods such as questionnaires, diaries, and participating in interviews. According to Pettersson et al. (2018), self-developed questionnaires are the most employed ones (53%), followed by semi-structured interviews (46%) and standardized questionnaires (26%), such as AttrakDiff and the Self-Assessment Manikin (SAM) (Pettersson et al., 2018). A drawback of this approach is that it relies on retrospective evaluation. Users may find it difficult to remember every detail of their experiences. By contrast, the events they report in the questionnaires or interviews may reflect the most important events during their interaction.

Another approach that is being used more recently, also based on self-reporting, is the analysis of online user reviews. These reviews are spontaneously written by users and are sources of experience reports with a product (Pagano and Maalej, 2013), which can be used to measure quality attributes (Hedegaard and Simonsen, 2013). In this approach, a massive amount of data is obtained from reviews from social networks, such as Twitter and Facebook, as well as websites and app stores. These data are then analyzed by employing machine learning techniques or manually extracting information that helps developers better understand users' opinions about their product or from their competitors.

This feedback from users is very important for software development companies, given that more than 70% of people read reviews before downloading an app, and 75% consider them as a key driver for downloading, being considered even more trustworthy than personal recommendations by 42% of people (Weichbroth and Baj-Rogowska, 2019). Developers can use the information obtained through these reviews to prioritize development efforts, either for an app refinement or competitive market entry perspective (Simmons and Hoon, 2016). Moreover, negative reviews can significantly influence the company's revenue and product awareness rate (Hoon et al., 2012). Figure 2.1 presents a review written by a user about the WhatsApp application in Google Play Store. He complains that the app does not automatically split a long video into smaller ones. Additionally, he states that the app is developed by the same company that maintains Instagram and Facebook, which have this feature available, thus calling the development team lazy. The user ended up giving just one star in his review. Moreover, other 253 people liked this review, indicating that they may have faced similar problems and share the same opinion. It indicates that many people read these reviews, raising the importance of considering this source of information for app development and improvement.



[redacted]

★☆☆☆☆ July 15, 2020

253

Its really annoying having to take so much time to figure out how to post a long video in status. The app should be able to separate a 1 minute video into two 30 second videos like on instagram. I dont like having to do that manually...the video ends up not flowing. I love the app but i just hate this so much...and since whatsapp and instagram are both under Facebook i feel like youre just being lazy

Figure 2.1 - Example of a user review about WhatsApp from the Google Play Store.

2.3. RELATED WORK

Although many studies explore user reviews from app stores, five works are the most similar to our proposal. We present these works below.

Hedegaard and Simonsen (2014) proposed a tool to extract usability and UX (UUX) information from online user reviews. Their goal was to investigate the amount of UUX information that can be obtained from these reviews. This tool uses a machine-learning classifier that automatically tags sentences in reviews according to the information related to usability or UX dimensions. In this work, the authors selected UX dimensions from six different works (Bargas-Avila and Hornbæk, 2011; Bevan, 2008, 2009; Folmer et al., 2003; Ketola and Roto, 2009; Seffah et al., 2006)and categorized them into four groups as follows: CLASSICUA, Bevan, Ketola, and Frequent. CLASSICUA refers to classic usability dimensions: memorability, learnability, efficiency, errors/effectiveness, and satisfaction. There are the following dimensions In the Bevan’s set: likeability, pleasure, comfort, and trust. The Ketola’s set comprises: anticipation, overall usability (i.e., whether the user was successful using the product), hedonic, detailed usability (i.e., functions used, usability problems, and performance), user differences, support, and impact. Finally, Frequent refers to frequently mentioned dimensions: affect and emotion, enjoyment, fun, aesthetics, appeal, engagement, flow, motivation, enchantment, frustration, and hedonic. The authors focused on the software and video games category from the epinions.com website. They performed a manual annotation of 6,655 sentences, which served as input to train the classifier. They employed the TF-IDF approach and the Support Vector Machine (SVM) classifier. They extracted the most informative words by selecting those with the largest distance to the hyperplane afforded by SVM. The authors identified that classic usability measures are more frequent in the software category, while video-game reviews emphasize dimensions related to emotions.

Bakiu and Guzman (2017) presented an approach to automatically extract software features from app store reviews and visualize users’ satisfaction with these features regarding the UUX dimensions from the work of Hedegaard and Simonsen (2013). The tool uses Part-

Of-Speech (POS) tagging and collocation finding algorithm from the Natural Language Toolkit (NLTK) platform[3] to extract features. The tool also uses SentiStrength to estimate users' satisfaction. Finally, a machine learning classifier is used to automatically detect and classify specific UUX information associated with each extracted feature. The authors trained and tested the classifier with the same sets of reviews from the work of Hedegaard and Simonsen (2013). They also created a golden standard of manually labeled sentences to compare the results of their approach. The results were mixed among the dimensions, with better results for four out of 14 dimensions that appeared more than 15 times in the test set (Satisfaction; Engagement and Flow; Detailed Usability; Aesthetics and Appeal).

In contrast to these works from Hedegaard and Simonsen (2014) and Bakiu and Guzman (2017), 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 process. Their proposal also has overlapping dimensions, such as "Hedonic", "Pleasure", and "Affect and Emotion". The UX dimensions they selected are mainly focused on emotions and feelings. Although it is important to assess users' sentiments, the lack of classification regarding functional aspects and features of an app makes it difficult for the developer to identify, for instance, what new features users are requesting and what is their opinion about a recent update.

Jang and Yi (2017) extracted four UX aspects (expectation confirmation, hedonic, pragmatic, and user burden) from online user reviews and analyzed their impact on users' overall satisfaction (i.e., the star rating of each user). They applied Linguistic Inquiry and Word Count (LIWC) 2015 dictionary (Pennebaker et al., 2015) to extract the UX aspects and performed a linear regression to analyze their impact on users' satisfaction. The authors found mixed results. Hedonic aspects (positive emotions) positively affected user satisfaction, while user burden had a negative effect. The effect of pragmatic aspects, in turn, varied according to the context (work, home, or leisure). Finally, expectation confirmation had a significant effect only for smart TVs. The main limitation of this work is that it does not employ Natural Language Processing (NLP) or Machine Learning (ML) techniques to identify and extract the UX aspects, but the LIWC tool, which considers only a single keyword to analyze and identify them. In our proposal, we employed both NLP and ML techniques to analyze and extract factors from user reviews. Moreover, the focus of our research is different. While Jang and Yi (2017) analyzed user reviews of products from three categories from Amazon.com, we focused on user

[3] http://www.nltk.org

reviews from app stores, which have specificities, such as the unstructured nature, proper punctuation, and short length in comparison to general product reviews (Fu et al., 2013).

McIlroy et al. (2016) proposed a multilabel approach to classify user reviews. To do so, they manually analyzed a sample of reviews with 1 and 2 stars and identified 14 factors: Additional cost, Functional complaint, Compatibility issue, Crashing, Feature removal, Feature request, Network problem, Privacy and ethical issue, Resource heavy, Response time, Uninteresting content, Update issue, User interface, and Other. Then, they trained an ML model by applying Naïve-Bayes, Decision Tree, and SVM classifiers. Due to the low performance of Response time, Uninteresting content, and User interface, they merged these factors into the Other factor. The authors presented three scenarios to validate their proposal: i) app comparison; ii) app store overview; and iii) anomaly detection. This is the closest work to our proposal. One of the differentials of our research is in the methodology employed to define the set of factors. While they performed qualitative analysis on a sample of reviews, we performed a systematic mapping study from several works to have a broader coverage of influencing factors. Their proposal was also focused only on negative reviews and factors, being not suitable, for instance, in scenarios where a company wants to identify the most important features from a competing app. Finally, our proposal focuses not only on classifying the reviews and presenting the distribution of positively, negatively, or neutrally evaluated factors but on providing a set of top features that developers should consider when developing new applications or improving existing ones.

2.3.1. Research Opportunities

Based on the results from the studies presented above, we identified the following gaps and limitations:

- Some emotions, such as sadness, although negative in nature, did not negatively influence users' evaluations in the context of games. This highlights that some emotions can affect users' evaluations in a different way according to the type of product under evaluation;
- Some studies were conducted in specific conditions, such as using applications designed specifically to induce a higher effect of the evaluated phenomenon through limited interaction or in particular types of applications, such as games and social networks. There is a need for further studies investigating the impact of these factors during actual users' interaction with real applications, as well as in other contexts

and types of applications to make it possible to identify the specificities of each scenario;

- Existing approaches are not designed for app store reviews, consider only negative reviews, or have little focus on pragmatic aspects of the experience. The factors these approaches consider are also limited by the dataset extracted during their development, which reduces its scope. Moreover, these proposals were not evaluated from the practitioners' perspective, being assessed only through classification metrics, such as precision, recall, and F1-score. There is a need for an approach that has a more holistic view of the factors affecting UX and provides valuable and relevant results from practitioners' perspectives.

2.4. SUMMARY

In this chapter, we presented the concept of UX and the results of some studies that investigated the factors that can influence the perceptions of the experience, such as the peak-end rule, previous experience, and the number of problems. However, further studies are necessary to investigate their effects. Regarding the peak-end rule, for instance, Gutwin et al. (2016) found that not all predicted effects were confirmed. This factor had mixed effects according to the type of game. The authors also suggest that further studies are necessary to generalize the results by analyzing different types of games and manipulating other variables. It is needed to investigate whether the peak-end rule applies to our context, i.e., mobile apps.

We also presented different approaches to evaluating UX. Among them, the analysis of self-reported data provided by the crowd, such as user reviews from app stores, has gained importance in recent years. The analysis of these reviews can bring valuable information to understand users' needs, bugs, and improvement opportunities. Different from studies conducted in lab settings, users provide such feedback spontaneously, expressing what they loved and hated the most in the app. Moreover, the variety of reviews for different types of applications results in a huge amount of data that would be hard to collect from empirical studies, making it a valuable source for analyzing influencing factors.

CHAPTER 3 – INVESTIGATING INFLUENCING FACTORS

This chapter presents our first iteration over the relevance cycle. We carried out an empirical study to investigate whether different factors influence users' perception of the experience. The results of this study allowed us to identify which factors affect users' evaluations and to what extent.

3.1. INTRODUCTION

In this chapter, we present the first iteration of the relevance cycle. We aimed to investigate why users keep evaluating their UX as positive even when facing many usability problems and expressing negative emotions during their interaction with a software product. This phenomenon was already noticed by Hassenzahl (2018a), leading him to propose one of the most influential models of UX that separates the pragmatic aspects related to task accomplishment from the hedonic aspects related to user's emotions and affect (Hassenzahl, 2007).

Although this conceptual model might explain this divergence, the results from one of the previous studies of our research group (Nakamura et al., 2020) indicated the influence of other factors on users' perceptions of their experience, such as the nature of the method employed, as well as users' profile. In that study, we evaluated the UX of a web platform designed for a government traffic department. We divided the participants into two groups: one group comprised experts in HCI, who acted as inspectors, and the other group comprised company employees not related to the software development, who acted as users in a testing session. The results indicated that experts considered their experience neutral, while the employees evaluated their experience as very positive. The lower ratings from inspectors raised the possibility of the influence of the inspection process, which requires them to detect as many problems as possible, leading them to focus on the negative aspects of the platform and influencing their evaluations. By contrast, the higher ratings from the employees may be related to their profile. Given that they use computers only occasionally, they probably were not familiar with this type of platform. Thus, they had no expectations about it nor any previous experience to compare to, which may have resulted in more positive ratings. In this sense, it is essential to investigate these factors to understand UX better and progress the research in the field.

In the last decades, researchers have been focusing their efforts on understanding how users form their judgments about their experience when interacting with a product. Some researchers investigated whether aspects related to user interaction affect their evaluations. Hassenzahl et al. (2002), for instance, investigated the influence of hedonic and pragmatic attributes on appeal according to how users use the product. They found that people using a given product in goal-mode, i.e., by giving them tasks to be accomplished, tended to evaluate it according to its capability of supporting goal achievement, focusing on pragmatic aspects. By contrast, pragmatic aspects became less important for people in action-mode, i.e., when they are instructed to do what they like. In another work, Hassenzahl (2004) investigated the relationship between beauty, goodness, and usability. The results indicated that pragmatic and goodness were affected by usability problems, while hedonic attributes and beauty remained stable. In turn, Cockburn et al. (2017) evaluated the influence of the peak-end rule by manipulating the interaction sequencing. To do so, they examined user preferences for a series of interactions with different orderings to create positive and negative recency and primacy effects. Primacy refers to the over-weighted influence of the first experience, which has a substantial and lasting effect on the participant's subsequent behavior (Shteingart et al., 2013). On the other hand, recency is when the latest experience is more influential on the participant's judgments (Hands and Avons, 2001). The results indicated a significant influence of recency effects on users' perceptions about their experience with interfaces that, otherwise, are identical in their objective interaction requirements. By contrast, the influence of the primacy effect was not observed.

Other works focused on users' aspects, such as previous experience. The study from Langdon et al. (2007) with a digital camera and a car indicated that prior experience with similar products is a strong predictor of the usability of products. People without experience resorted to a trial and error-based approach to accomplish the tasks, leading to a slower, more repetitive, and error-prone interaction. In another work, Sagnier et al. (2020) found that participants with prior experience in Virtual Reality gave significantly higher scores for pragmatic quality than those without experience. By contrast, participants without experience evaluated the hedonic quality stimulation with significantly higher ratings. Finally, Schneidermeier et al. (2014) investigated the effect of changing paradigms. They evaluated the UX of Windows 7 and Windows 8 operating systems by dividing the participants into three groups: Windows 7 users, Windows 8 users, and no Windows user. Microsoft changed some parts of the interface from one version to the other. In Windows 8, the Start menu was removed from the taskbar and placed into a hidden menu that can be accessed when pointing the mouse to one of the corners on the right of the screen. The results indicated that Windows 7 was perceived as more task-oriented, while Windows 8 was neutral. According to the authors, the neutral score may be due to disagreements about the pragmatics of the system.

Although these works provided insights regarding the possible influence of factors such as the number of problems (Nakamura et al., 2020), previous experience (Sagnier et al., 2020), and interaction sequencing (Cockburn et al., 2017), some gaps remain open, requiring further studies to investigate the influence of these factors. Regarding the influence of the number of problems and previous experience on UX, our previous work (Nakamura et al., 2020) indicated that inspectors evaluated the UX conveyed by the product significantly lower than users. The number of problems that inspectors identified during their tasks may have influenced their perceptions of the UX. However, the users' profiles may have contributed to this difference, given that they only use computers occasionally, thus, having low experience compared to inspectors. The effect of previous experience also remains unclear, especially when involving applications with innovative forms of interaction that change paradigms. Therefore, it is essential to investigate better the influence of the number of problems and previous experience on UX. Regarding interaction sequencing, although Cockburn et al. have found a significant influence in their works (Cockburn et al., 2017, 2015), Gutwin et al. (2016) obtained mixed results according to the type of game, which indicates that this factor may not have a substantial effect in some contexts. Thus, it is necessary to investigate whether this factor significantly influences users' perception of the experience in our context, i.e., mobile apps.

In this study, we evaluated a mobile app with a novel interaction approach with a chatbot to investigate the effect of these three factors. 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 a real 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.

By identifying which factors influence users' perception of their experiences, researchers may carry out more precise evaluations by designing the study to minimize the effect of these unwanted variables on the results. Moreover, practitioners may focus their efforts on factors that impact UX, making it possible to optimize the development process and provide a better experience for users. We present the details of this study in the following subsections.

3.2. METHODOLOGY

3.2.1. Participants and Materials

We conducted the study with 33 participants among inspectors and end-users. The inspection group comprised four men and seven women between 22 and 39 years old (average 30), all of them working on research in HCI and SE fields. Among them, nine were doctorate students, and two were master's students in informatics from the Federal University of Amazonas (UFAM). For the end-users group, we randomly selected the participants from the university's staff and students from various courses as we met them in the corridors. A total of 22 participants volunteered to participate in the test. Among them, 16 were men and 6 women, aged between 18 and 58 years old (average 27).

We used the following materials to carry out the study: i) an informed consent form (research project approved by the ethics committee of the Federal University of Amazonas - UFAM - Certificate of Presentation for Ethical Consideration–CAAE number 79972917.0.0000.5020); ii) a characterization questionnaire; iii) two scripts with the set of tasks; vi) smartphones.

3.2.2. UX Evaluation Methods

As our goal with the stakeholders is to identify as many problems as possible, we employed methods that use different approaches in this study: inspection and testing. We also employed a questionnaire-based method to quantitatively assess the quality of the experience conveyed by the application and identify which aspects of UX can be improved.

Despite being employed for more than three decades to identify problems in usability evaluations (Lallemand et al., 2014) and commonly being used in the industry (Fernandez et al., 2013), inspection is not very common to evaluate UX. It is because UX evaluations relies on the observation of users performing a set of tasks while interacting with a given product (Lallemand et al., 2014). Inspections, in turn, consist of the evaluation of an interface by reviewing a set of principles named heuristics to detect problems that may affect user’s interaction, being usually performed by experts (Johannessen and Hornbæk, 2014). A positive aspect of this approach is that it is cost-effective, as it does not require users to conduct evaluations, which takes a lot of time and effort (Alves et al., 2014). The Inspection also does not require any special equipment, in addition to detecting a wide range of possible faults in a limited amount of time (Matera et al., 2002). Companies are constantly looking for ways to reduce costs, so it may be a good alternative to identify problems.

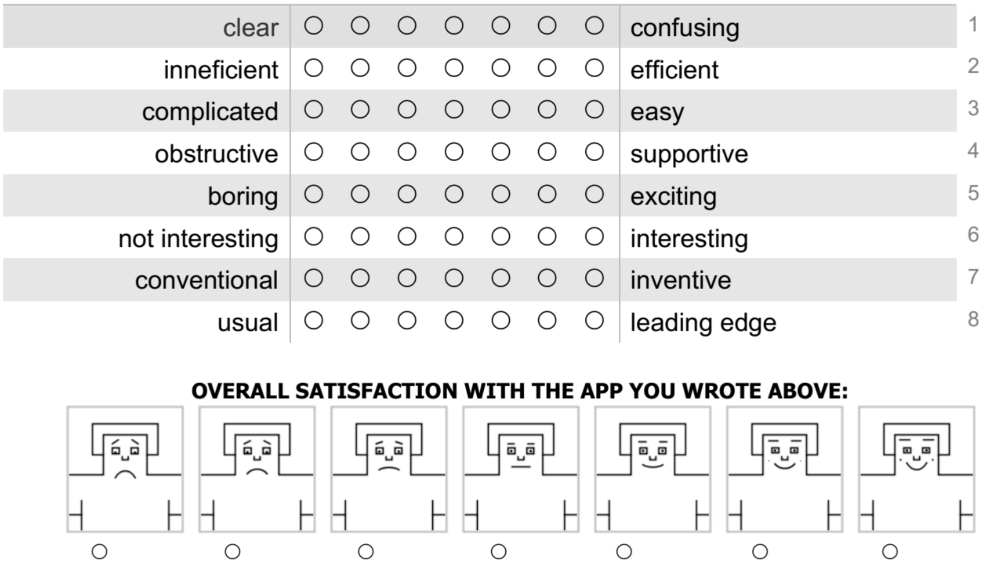
The inspection group applied a UX inspection method called UX-Tips (Marques, 2019). Unlike traditional usability inspection methods in which the set of heuristics focuses on evaluating pragmatic aspects, UX-Tips also focuses on hedonic aspects. Overall, the method evaluates the following set of factors that the authors obtained from a literature review, each factor with a set of heuristics: aesthetics, emotion, engagement, innovation, social, physical features (for mobile devices), ease of use, and learnability, utility, control, feedback, efficiency, added value, and satisfaction. A previous study (Marques, 2019) has demonstrated that UX-Tips was more effective and efficient than another UX inspection method from the literature, allowing the identification of a greater number of problems with fewer false-positives. Moreover, inspectors who employed UX-Tips also provided, in their reports, details about how the problem affected their experience with the evaluated application. Such behavior, however, rarely occurred in the reports from the inspectors that employed the other method, whose content was limited to just describing the problem they identified. Details of the UX-Tips method can be found in ANNEX C.

For testing, we looked for methods that: (i) are easy to be applied; (ii) do not require additional equipment (e.g., eye-tracking devices); (iii) are not much time consuming; (iv) require not more than one observer per participant; and (v) provides real-time information without obstructing participant's interaction with the platform. Considering these criteria, we selected Concurrent Think-Aloud (CTA). CTA is one of the most widely used testing methods and allows the detection of a high number of problems with less time than other similar methods (Alhadreti and Mayhew, 2018). CTA is a variation of the Think-Aloud method that provides "real-time" information during the participant's interaction with a system (Alhadreti and Mayhew, 2018). The participant performs tasks as they verbalize their thoughts while being observed by a moderator that takes notes about the participant's interaction in a problem reporting form, making it easier to identify where and what causes the problem. The problem can be identified through three approaches (Alhadreti and Mayhew, 2018): i) observation (i.e., from observed evidence without verbal data); ii) verbalization (i.e., from verbal data without accompanying behavioral evidence); and iii) combination of observation and verbalization.

Finally, to complement the results of both methods with an overall measure of the UX from users' points of view, we looked for a quantitative UX evaluation method. To do so, we analyzed the methods identified in the systematic mapping conducted by Rivero and Conte

(Rivero and Conte, 2017) as a starting point. We defined criteria based on eight research sub-questions (SQ) expressed by the authors as follows. We excluded methods that: (i) EC1: do not obtain data directly from users (SQ2); (ii) EC2: cannot be applied in controlled environments (SQ3); (iii) EC3: were not applied to evaluate mobile applications (SQ4); and (iv) EC4: are not available for download/ consultation (SQ8). From the 227 publications, 222 did not meet the criteria and were excluded, resulting in five publications with nine unique methods. From this set, we performed further refinements. Some methods were generic (e.g., interviews and observation), while others required specific equipment, such as sensors and electroencephalogram, or focused on specific variables, such as effort. We ended up with two UX evaluation methods analyzed in detail: AttrakDiff (Hassenzahl et al., 2003) and User Experience Questionnaire (UEQ) (Laugwitz et al., 2008). To decide which method to choose, we examined their feasibility and comprehensiveness. A study from Marques et al. (2018) indicated that AttrakDiff uses technical terms that are not easy to understand, leading participants to answer anyway and impacting the evaluation results. In turn, UEQ was perceived as very easy to use (Nakamura et al., 2019), while providing a tool to analyze the data from both short and complete versions of the method. It also surpassed AttrakDiff in 2017 in the number of studies employed (Díaz-Oreiro et al., 2019). Thus, we selected UEQ.

UEQ comprises 7-point semantic differential scales where the participant should mark the point closest to the adjective that better conveys his/her experience with the product. To avoid participants' fatigue and reduce the time required for user testing, we employed the shortened version of UEQ (Schrepp et al., 2017), which focuses on the two general UX dimensions: Pragmatic Quality (PQ) and Hedonic Quality (HQ), evaluated by four pairs of adjectives each. At the end of the evaluation, we asked the participants to assess their overall satisfaction with the mobile app through the Valence (pleasure) dimension of the Self-Assessment Manikin method (Bradley and Lang, 1994), given that this dimension reflects users' tendency to approach or withdraw from an experience. It consists of a visual evaluation of the experience, from images ranging from an unsatisfied to a satisfied face and a semantic differential scale. We adopted the 7-point scale version to be comparable with the results from UEQ. Figure 3.1 presents the short version of UEQ method with the valence dimension from SAM.



clear	○	○	○	○	○	○	○	confusing	1
inneficient	○	○	○	○	○	○	○	efficient	2
complicated	○	○	○	○	○	○	○	easy	3
obstructive	○	○	○	○	○	○	○	supportive	4
boring	○	○	○	○	○	○	○	exciting	5
not interesting	○	○	○	○	○	○	○	interesting	6
conventional	○	○	○	○	○	○	○	inventive	7
usual	○	○	○	○	○	○	○	leading edge	8

OVERALL SATISFACTION WITH THE APP YOU WROTE ABOVE:

○ ○ ○ ○ ○ ○ ○

Figure 3.1 - Short version of UEQ added with SAM's valence dimension.

It is worth mentioning that we adapted UEQ to Brazilian Portuguese, as it only has the European Portuguese version. All the process was carried out by three researchers and reviewed by a senior researcher expert in HCI. The adaptation may impact the reliability of the instrument, given that the words may have a different meaning when translated. Thus, in order to assess the reliability of the translated version of UEQ, we calculated Cronbach's Alpha (Cronbach, 1951). The results indicated a high degree of internal consistency for both PQ and HQ dimensions, with Cronbach's alpha coefficients of .891 and .863, respectively. We also carried out a Principal Component Analysis with Varimax rotation, as employed by the authors of UEQ (Schrepp et al., 2017), to check factor loadings and evaluate construct validity (Williams et al., 2010). The analysis extracted two factors explaining 76.7% of the variance, producing the expected pattern as output (see Table 3.1). The only pair of adjectives with a relevant cross-loading is "boring/exciting." It is possible that these adjectives do not fit well in the context of the application, making the participants not to be sure about their meaning or how to evaluate them.

3.2.3. Test Object and Tasks

The test object of this study was an app designed to facilitate shopping in local markets. It is developed by a local software development company and the stakeholders wanted us to help them attract more consumers by improving the app's quality aiming to provide a better user experience. In this app, users interact with a chatbot through the options that it provides or by typing a text like in a chat, making the buying process more informal and interactive compared to traditional shopping apps.

Table 3.1. Factor loadings

Item	PQ	HQ
confusing / clear	**.785**	.031
inefficient / efficient	**.800**	.159
complicated / easy	**.905**	.197
obstructive / supportive	**.882**	.229
boring / exciting	.553	**.602**
not interesting / interesting	.468	**.823**
conventional / inventive	.049	**.877**
usual / leading edge	.151	**.878**

We built two scripts with tasks to investigate the interaction sequencing effect for this study. We designed one script to simulate the negative beginning and positive ending condition (+*end condition*) and another to simulate the positive beginning and negative ending condition *(-end condition*). To achieve this, we ordered the tasks in two different forms. Both scripts start with account creation. However, the other three tasks are in the opposite order in each script. In the +*end condition* (see Table 3.2), we ordered the tasks so that the participants started with a time-consuming task in which they needed to interact with the chatbot several times to add six different products required in the script. The second task requires only one product to be added and to be modified next. Finally, the last task requires only one product to be added. The -*end condition* has exactly the same tasks, but with Tasks 2, 3, and 4 in the opposite order. All the tasks involved an interaction with the chatbot, except when searching for the stores (through the search bar on the top of the app) and looking for whether the store delivers to the specified address.

3.2.4. Procedure

We conducted the inspection and testing on different days in a research laboratory at the Federal University of Amazonas (UFAM). Before the evaluation process, we asked the participants to review and sign an informed consent form. They also filled in the characterization questionnaire, which was used to divide them into two groups, each assigned to one script and balanced according to: i) experience with usability/UX evaluations (only for the inspection group); ii) prior use of similar apps; iii) shopping apps usage frequency; iv) prior use of the target application. Table 3.3 and Table 3.4 present the profile of each participant per group.

Table 3.2. Set of tasks for the +*end condition*.

#	Task description
1	Create an account in the app
2	Let’s place the monthly order! - Go to the XYZ store - Check if they deliver to 123 road - Add the following items into your cart (6 products from the XYZ store) - Remove the following item from your cart (1 item added earlier) - Add the following item in your cart (1 new item) - Do the checkout procedure - On the payment screen, cancel the order
3	Let’s order a hot dog! - Go to the WWW store - Select Classical Hot Dog - Choose some extras - Do the checkout procedure - On the payment screen, cancel the order
4	Let’s buy a new TV! - Buy a new TV of any brand and model - On the payment screen, cancel the order

For experience with usability/UX evaluations (**Evaluation Exp.**), we considered the following: a) *None:* never heard about usability/UX evaluation; b) *Low:* have already read about usability/UX evaluation before, but not in-depth; c) *Medium:* have learned about it in classes or courses and have performed exercises in classroom; d) *High:* have already performed/participated in usability/UX evaluations. None of the participants had previous experience with UX-Tips. The options were binary for previous experience with similar apps (**Similar app?**): Yes or No. For similar apps usage frequency (**Usage frequency**), we classified them as follows: a) *None (N):* does not use this type of app for a very long time; b) *Low (L):* usually uses once a month; c) *Medium (M):* usually uses once a week; d) *High (H):* uses many times a week. Finally, for previous use of the target app (**Target app**), the option was also binary: Yes or No.

Table 3.3. Overview of inspection groups’ profile.

Inspection Group 1 (+*end condition*)						
ID	**I2**	**I3**	**I4**	**I5**	**I8**	**I9**
Evaluation Exp.	High	High	None	High	Medium	High
Similar app?	Yes	Yes	No	Yes	Yes	Yes
Usage frequency	None	Medium	None	Medium	None	Low
Target app?	No	Yes	No	No	No	Yes
Inspection Group 2 (-*end condition*)						
ID	**I1**	**I6**	**I7**	**I10**	**I11**	
Evaluation Exp.	High	High	High	High	High	
Similar app?	Yes	No	Yes	Yes	Yes	
Usage frequency	None	Low	Medium	Medium	None	
Target app?	No	No	No	Yes	No	

Two researchers conducted the inspection phase with 11 participants who acted as inspectors. After signing the consent form and filling the characterization questionnaire, we introduced them to the UX-Tips method, explaining what it was designed for and how to use and report problems with it, without giving much details. We also explained the purpose of the target application and provided the script with the set of tasks to be performed during the inspection process according to the condition the inspector was assigned to. Each participant completed the inspection individually in smartphones we provided without the interference of the researchers. Finally, at the end of the inspection, the inspectors filled in the shortened version of UEQ (Schrepp et al., 2017). We also asked those who already used similar apps to rate the UX of a similar application that they remember before evaluating the target application to understand the relationship between previous and current experience better (see ANNEX D to visualize the full questionnaire).

Table 3.4. Overview of the participants who acted as users in usability testing

Testing Group 1 (+*end condition*)											
ID	**U5**	**U6**	**U7**	**U8**	**U11**	**U12**	**U14**	**U16**	**U17**	**U18**	**U19**
Similar app?	Yes	No	Yes	No	Yes	No	Yes	No	No	No	Yes
Usage frequency	N	N	N	N	N	N	M	L	N	N	N
Target app?	No	No	No	No	No	No	No	No	No	No	No
Testing Group 2 (-*end condition*)											
ID	**U1**	**U2**	**U3**	**U4**	**U9**	**U10**	**U13**	**U15**	**U20**	**U21**	**U22**
Similar app?	No	No	No	No	Yes	No	Yes	Yes	Yes	Yes	No
Usage frequency	N	N	N	N	M	N	M	L	L	M	L
Target app?	No	No	No	No	No	No	No	No	No	No	No

The same two researchers conducted user testing with 22 other participants who acted as end-users. A designer of the target application and two developers, undergraduate students from UFAM enrolled in a Computer Science course, acted as moderators together with the researchers. The two undergraduate students had taken Human-Computer Interaction classes and had already performed usability evaluation exercises in the classroom. The designer also had experience with this type of evaluation in the industry. Each moderator conducted the test with one participant at a time. First, they introduced the evaluated application goals. Then, they provided the script with the set of tasks the participant should perform (Table 3.2), according to the condition (+*end* or -*end*) they were randomly assigned to. They also asked the participants to verbalize their thoughts during their interaction. The moderator was responsible for taking notes of any interaction issues identified through the participant's verbalization or from their own observation. To ensure the rigor of their evaluations, one of the researchers first conducted a testing session with a participant while they observed the process. After, each of them conducted their evaluations with the participants individually, while the researchers observed and checked their annotations during the first sessions.

The study took place in a research lab of UFAM. Each participant was randomly assigned to one condition (*-end* or *+end*). First, the moderator briefly described the evaluation process and introduced the application to the participant, explaining what it was designed for. Next, the moderator presented the set of tasks that the participant should perform in the application, being asked to verbalize their thoughts during the interaction. Each moderator conducted the test individually with one participant at a time, observing the participant and taking notes. At the end of the interaction, the participant filled in the UEQ. As with the inspection group, we asked participants who already used similar apps to rate first the UX of a similar application that they remember.

After conducting the two phases, we started the problem extraction and consolidation process. We divided the extraction process between two researchers. The process consisted in reading the description of each discrepancy, i.e., every potential problem reported by the participants, and extracting the main problem from it. Each discrepancy was classified into a problem (issue that should be fixed), false-positive (issue that did not represent a real problem in the app), or duplicate (discrepancy that are related to the same main problem). One researcher extracted the problems, and another researcher reviewed them. Divergencies between them were solved through a discussion session.

Finally, we generated a report and presented the stakeholders with the main problems identified (those with the highest number of occurrences). Based on the results of this study, the developers redesigned the application by considering the points highlighted in the report.

3.3. RESULTS

This section presents the results of the study. We divided it into two main subsections, according to the data being analyzed: i) **Inspection and Testing:** related to the comparison between the two approaches; and ii) **UX Evaluation (UEQ):** where we present the results regarding the investigation of the factors that may influence on users' perception about their experiences and reflect on their ratings.

During the analysis, we performed some statistical tests according to the distribution of the data. As the number of participants is below 50, we performed the Shapiro-Wilk normality test (Shapiro and Francia, 1972). If *p-value* > 0.05 (i.e., the data distribution is normal) in both groups for a given indicator, we applied Student's parametric t-test for independent samples

(Wohlin et al., 2012). Otherwise, if *p-value* < 0.05 (i.e., the data does not follow a normal distribution) in at least one group for that indicator, we applied Mann-Whitney non-parametric statistical test (Wohlin et al., 2012). Additionally, we calculated the effect size I at a 95% confidence interval by using Cohen's *d* (Cohen, 1988) according to Fritz et al. (2012) to measure the magnitude of the treatment effect in the cases where statistical significance was obtained.

3.3.1. Inspection and Testing

To compare the results from both evaluation methods, first, we accounted for the number of problems identified by each participant. Then, we compared the results regarding effectiveness and efficiency on identifying problems. We selected these indicators to reflect aspects that companies with budget and time constraints may consider when choosing a method. According to Ardito et al. (2014), one of the most reported problems by practitioners regarding usability/UX evaluations is that this type of evaluation requires many resources in terms of cost, time, and involved people. In this sense, the selected method must address as many problems as possible (effectiveness) with less time (efficiency) while requiring as few people as possible to reduce costs.

For **efficiency**, as each method requires a different number of persons to be employed, we calculated the person-hour efficiency. While inspection methods require only the inspector to be employed, testing requires at least two persons: the moderator who observes the participant and takes notes, and the participant themselves, as shown in Figure 3.2. Knowing the cost of employing such methods is very important for the industry when analyzing an adequate method for their needs. In this sense, we defined the following formula: $Efficiency_i = P_i / (time_i * n)$, where P_i and $time_i$ refer respectively to the total number of problems found by participant *i* and the time s/he spent in the evaluation (in hours), while *n* is the minimum number of people required to employ the method. The result is the number of problems per hour that the participant can find. We defined **effectiveness** as the ratio between the number of problems identified by the participant and the total of all problems identified. We formulated the following null and alternative hypotheses:

H_1: There is no difference in effectiveness between inspection and testing.

H_{A1}: There is a significant difference in effectiveness between inspection and testing.

H_2: There is no difference in efficiency between inspection and testing.

H_{A2}: There is a significant difference in efficiency between inspection and testing.

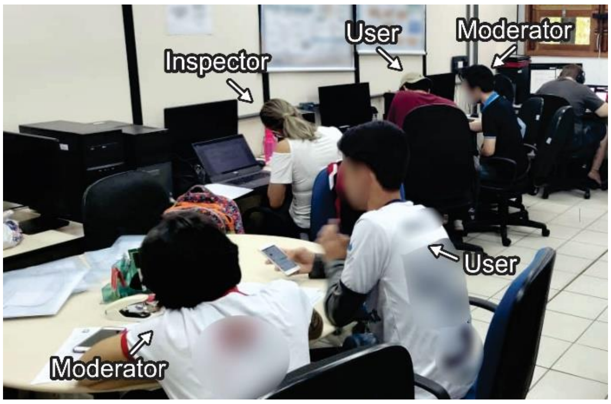


Figure 3.2. Experiment being conducted at a major lab.

Table 3.5. Results from the inspection group.

	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	I11
Discrepancies	13	7	27	21	26	18	27	9	30	27	10
Repeated	0	0	1	0	3	1	5	0	5	0	0
False-positives	0	0	0	0	1	0	0	0	0	2	2
Problems	13	7	27	21	26	18	27	9	30	27	10
- Unique	7	5	21	11	9	6	7	1	12	10	5
- Duplicated	6	2	5	10	13	11	15	8	13	15	3
Time (hours)	1.10	1.20	1.27	1.03	1.98	1.33	1.02	0.78	1.50	0.68	0.83
Effectiveness (%)	7.4	4.0	14.9	12.0	12.6	9.7	12.6	5.1	14.3	14.3	4.6
Efficiency	11.8	5.8	21.3	20.3	13.1	13.5	26.6	11.5	20.0	39.5	12.0
Effectiveness ($\bar{X}$)						10.1%					
Efficiency ($\bar{X}$)						16.2					

Table 3.6. Results from the testing group.

	U1	U2	U3	U4	U5	U6	U7	U8	U9	U10	U11
Discrepancies	6	13	9	9	7	10	7	11	7	2	7
Repeated	0	0	0	0	0	0	0	0	0	0	0
Problems	6	13	9	9	7	10	7	11	7	2	7
- Unique	5	10	4	7	3	4	5	1	3	0	1
- Duplicated	1	3	5	2	4	6	2	10	4	2	6
Time (hours)	0.30	0.32	0.50	0.33	0.30	0.30	0.45	0.50	0.23	0.12	0.22
Effectiveness (%)	3.4	7.4	5.1	5.1	4.0	5.7	4.0	6.3	4.0	1.1	4.0
Efficiency	10.0	20.5	9.0	13.5	11.7	16.7	7.8	11.0	15.0	8.6	16.2
	U12	**U13**	**U14**	**U15**	**U16**	**U17**	**U18**	**U19**	**U20**	**U21**	**U22**
Discrepancies	22	6	12	18	14	10	10	7	16	2	6
Repeated	1	0	0	1	2	0	0	0	2	0	0
Problems	21	6	12	17	12	10	10	7	14	2	6
- Unique	10	3	2	7	3	2	4	1	4	0	2
- Duplicated	11	3	10	10	9	8	6	6	10	2	4
Time (hours)	0.57	0.32	0.32	0.35	0.48	0.27	0.47	0.30	0.38	0.17	0.25
Effectiveness (%)	12.0	3.4	6.9	9.7	6.9	5.7	5.7	4.0	8.0	1.1	3.4
Efficiency	18.5	9.5	18.9	24.3	12.4	18.7	10.7	11.7	18.3	6.0	12.0
Effectiveness ($\bar{X}$)						5.3%					
Efficiency ($\bar{X}$)						13.7					

Table 3.5 and Table 3.6 present the data from the inspection and testing group, respectively. *Discrepancies* row refers to all potential problems reported in the problem reporting form. *Repeated* refers to discrepancies already reported by the same participant in different tasks of the script. *False-positives* are discrepancies reported by the inspector that are not real problems in the application. This row is not present in Table 3.6, as only problems faced by users are reported during user testing. *Problems* refer to validated discrepancies after the removal of false-positives and repeated ones, being categorized into unique (i.e., identified only by a given participant) and duplicated problems (i.e., already identified by other participants).

The results revealed a very low incidence of false-positives in the inspection group. Only 3 out of 11 inspectors reported discrepancies that are not real problems, indicating that the heuristics from the inspection method we employed (UX-Tips) is quite precise. The results also indicate that inspection performed better than testing regarding both effectiveness (10.1% vs 5.3%) and efficiency (16.2 vs 13.7 defects/hour). The statistical analysis showed that the effectiveness of UX-Tips was significantly higher than of CTA (t-test $t(13.860) = -3.515$, $p < 0.001$) with a medium effect size (Cohen's $d = 1.393$, $r = .571$), thus rejecting the H_1 null hypothesis. However, no difference in the efficiency was found (t-test $t(31) = -1.109$, $p = .276$), thus retaining the H_2 null hypothesis.

To investigate how well each method addressed the problems in the application, we analyzed the distribution of the problems identified (Table 3.7 and Table 3.8), and calculated two standard metrics proposed by Hartson et al. (2003): thoroughness and validity. **Thoroughness** is the number of problems identified that were confirmed in user testing (hits) divided by the total number of problems (hits + misses). **Validity**, in turn, is the number of problems identified and confirmed in user testing (hits) divided by the total number of problems identified in inspection (hits + false alarms). The formula to calculate the metrics is as follows:

$$thoroughness = \frac{hits}{(hits + misses)} \qquad validity = \frac{hits}{(hits + false\ alarms)}$$

A total of 175 unique problems (testing ∪ inspection) were addressed. Inspectors identified a greater number of these problems, addressing 73.7% of them, while user testing 45.1%. Inspectors also addressed 33 out of 79 problems that occurred during user testing, which gives us a thoroughness of 41.8% and validity of 25.6%. Although UX-Tips did not capture the remaining problems from user testing, it is worth mentioning that 8 out of 11 inspectors expressed their feelings about the application by selecting at least one item from a hedonic dimension (e.g., emotion, engagement) when reporting a problem. It allowed us to identify which aspect of the application led to negative sentiment about it. Inspector I4, for instance, reported: *"The application allows choosing products even if the store is closed. I chose several products, and when I went to the checkout, it informed me that I cannot checkout because the store is closed"*. To report this issue, the inspector selected the EMT2 item from the Emotion dimension, which has the following definition: "The application allows the user to feel happy using it". Inspector I10 also stated *"No. It's a lot of back and forth, especially when adding and removing items from the menu"*. To report this problem, the inspector selected the EMT1 item, which has the following statement: "It is pleasant / I like to use the application". In this sense, we could identify that they are potential problems that can lead users to have negative feelings about the application.

In contrast, despite being encouraged to verbalize their thoughts during their interaction, few users mentioned their experience during user testing. We identified only two reports where the sentiment was associated with the problem faced by the user in the following quotations:

"User found the bot cool, but a bit confusing" – Participant U2.

"Very annoying, it says to put the NIN [National Insurance Number], but it is not known where [to put it in]" – Participant U12.

Table 3.7. Distribution of problems identified by method.

Total	CTA (Testing)	UX-Tips (Inspection)
175	79	129
100%	45.1%	73.7%

Table 3.8. Unique and common problems found by each evaluation method.

	Testing	Inspection
Unique	46	96
Common	33	33
Subtotal	79	129
Total (Testing ∪ Inspection)	175	

3.3.2. UX Conveyed by the Application

We analyzed the results separately, according to our goals. First, we analyzed overall UX results. Then, we investigated whether previous experience and interaction sequencing influence UX evaluations. Finally, we carried out a correlation analysis to investigate the relationship between UX dimensions, number of problems, usage frequency, and overall satisfaction.

3.3.2.1. Evaluation Method and UX Evaluation

In an empirical study we conducted during the master's degree comparing inspection and testing methods (Nakamura et al., 2020), inspectors tended to provide significantly lower ratings for both Pragmatic Quality (PQ) and Hedonic Quality (HQ) dimensions than users. While the mean for each UX dimension for the testing group was positive, ranging between 1 and 2 (on a Likert scale from -3 to 3), the mean for the inspection was neutral, ranging from -1 to 1. It indicates a possible influence of the method on UX evaluations due to its focus on identifying problems. In this sense, we hypothesize that the number of problems identified by inspectors significantly affects their perception of both PQ and HQ dimensions. Thus, we defined the following hypotheses:

H_3: There is no difference in the ratings between inspectors and users for the PQ dimension.

H_{A3}: There is a significant difference between the ratings of inspectors and users for the PQ dimension.

H_4: There is no difference in the ratings between inspectors and users for the HQ dimension.

H_{A4}: There is a significant difference between the ratings of inspectors and users for the HQ dimension.

Figure 3.3 presents the distribution of the results for each of these items. The first four (clear, efficient, easy, and supportive) are related to the PQ dimension, while the next four (exciting, interesting, inventive, and leading-edge) are related to the HQ dimension. The last item is the Valence dimension from the SAM method (Bradley and Lang, 1994), which is related to overall satisfaction. To facilitate the analysis, we transformed the values ranging from 1 to 7 in the questionnaire into negative and positive values ranging from -3 to 3. Values above 1 indicate a positive perception of the UX conveyed by the application and values below -1 indicate negative perception (Santoso et al., 2016). Values between -1 and 1 suggest that the UX was neither perceived as positive nor as negative. For the PQ and HQ dimensions, we calculated the mean of the four items from each dimension per participant.

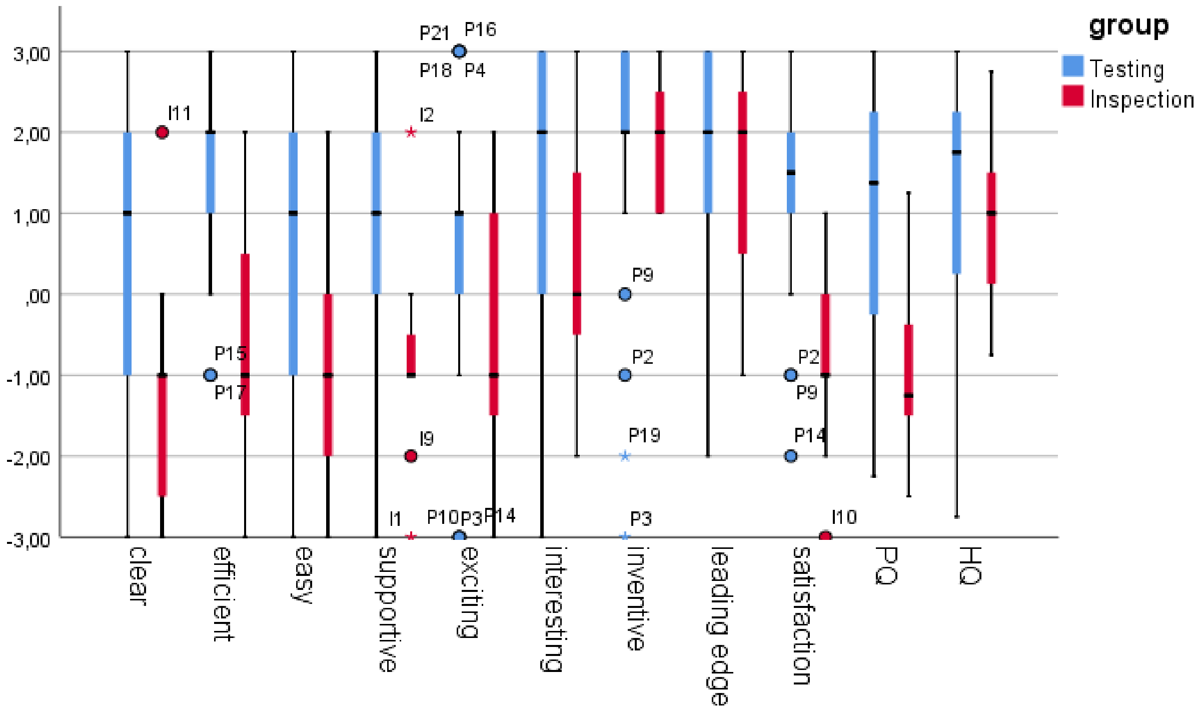


Figure 3.3. Overall UX evaluation by group.

The results revealed that the UX of the inspection group was contrasting with the testing group, precisely on the PQ dimension. While inspectors tended to have a more negative perception, users were more positive about their experience with the application, which was also reflected in their overall satisfaction. By contrast, the difference in the HQ dimension was smaller, and both groups tended to give more positive ratings, mainly on inventive and leading-edge adjectives, indicating that both groups recognize the innovative approach of the application.

The statistical analysis showed that the difference between the results from inspectors and users was statistically significant for PQ dimension with a large effect size (Mann-Whitney U = 34.5, z = -3.313, p < .001, r = .577), and overall satisfaction with also large effect size (U = 34, z = -3.405, p < .001, r = .593) with medium effect size (r = .593). perceived the app as not so straightforward and were less satisfied with the app than users. However, there was no significant difference in the HQ dimension between them (U = 81, z = -1.533, p = .125).

We also observed a greater number of outliers in the testing group. Probably it is because we analyzed all participants, with and without previous experience together. Thus, to investigate the influence of the method employed more consistently, we analyzed the data from both groups again, now by controlling for the previous experience variable. Given that only one participant from the inspection group did not have previous experience, we selected only the

19 participants who had already used similar applications (10 from the testing group and 9 from the inspection group).

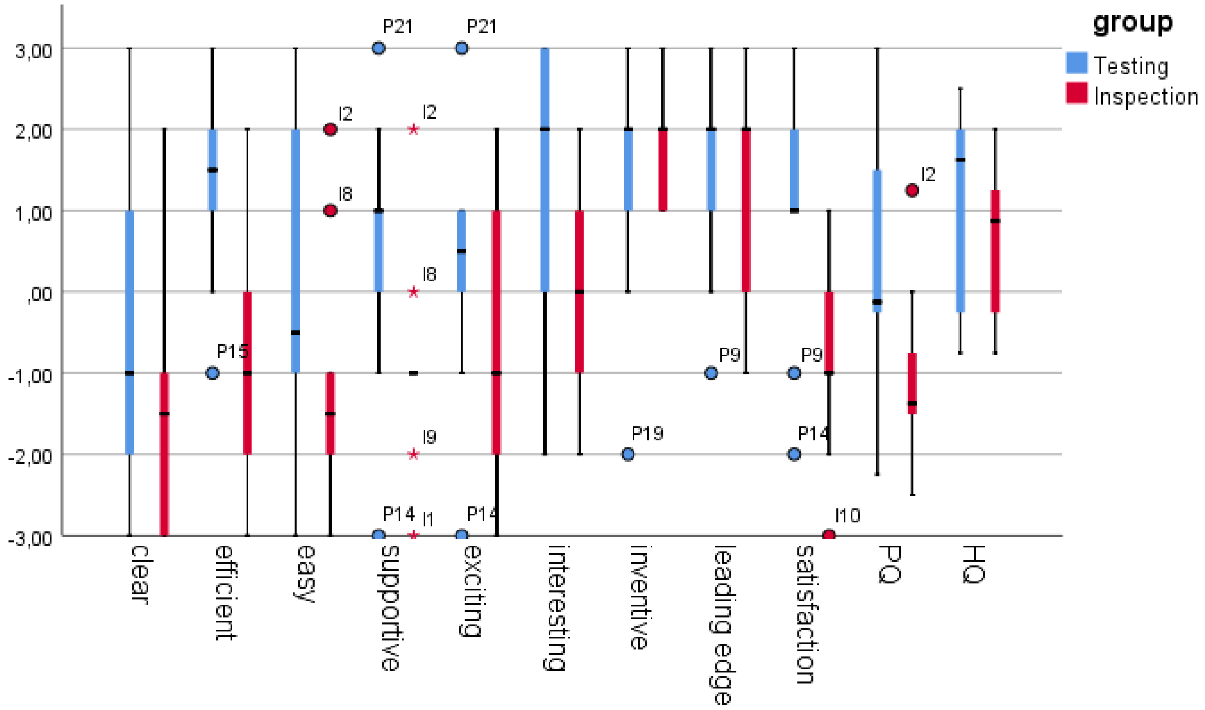


Figure 3.4. Comparison between UX evaluation of inspectors and users with previous experience.

Figure 3.4 presents the distribution of the evaluations from participants with previous experience with similar apps for each item and UX dimension per group. The results indicate that the perception of PQ items varied but of HQ was relatively similar. The statistical test revealed, in fact, that inspectors evaluated the PQ dimension and satisfaction significantly lower than users, both with medium effect size (*U* = 18.5, *z* = -2.175, *p* < .030, *r* = .499; *U* = 15, *z* = -2.514, *p* < .012, *r* = .494), thus rejecting the H_3 hypothesis. However, we did not find any statistical difference in HQ dimension (*U* = 30, *z* = -1.235, *p* = .217), thus retaining the H_4 null hypothesis.

To further analyze the influence of the method on UX evaluation and understand the reasons why inspectors rated the PQ dimension and overall satisfaction lower than users, we carried a Spearman’s correlation analysis between UX dimensions, number of problems, and overall satisfaction. Previous works have been pointing out that the mental effort caused by usability problems can affect the assessment of pragmatic quality while hedonic quality remains stable (Hassenzahl, 2004; Hassenzahl and Sandweg, 2004). Mental effort, together with technical problems and difficulty to use, were also the main reasons for unsatisfactory experiences with a mobile health application (Biduski et al., 2020). Given that inspection is focused on identifying as many problems as possible, our hypothesis is that the method can influence UX evaluations. As the number of problems inspectors identify increases, their perception of the app's usability and the experience with it may decrease, resulting in a negative correlation between number of problems, PQ dimension, and overall satisfaction. To better understand the results, we followed the interpretation for the field of psychology presented in Table 3.9 (Akoglu, 2018).

Table 3.9. Interpretation of the correlation coefficients.

Correlation Coefficient		**Interpretation**
Positive	**Negative**	
+1	-1	Perfect
+0.7 < *r* < +1	-0.7 < *r* < -1	Strong
+0.4 < *r* < +0.7	-0.4 < *r* < -0.7	Moderate
+0.1 < *r* < +0.4	-0.1 < *r* < -0.4	Weak
0	0	Zero

The results presented in Table 3.10 shows that the number of problems had a strong and moderate negative correlation with PQ dimension (*r* = -.712; *p* = .021) and satisfaction (*r* = -.634, *p* = .049) respectively. This indicates that the greater the number of problems the inspector identifies, the lower the rating s/he gives to PQ items. As the goal of the inspection is to find as many problems as possible, this focus on problems might have affected their perception of the app under evaluation, indicating that the method can significantly influence the results of UX evaluations.

Table 3.10. Correlation analysis for the inspection group.

n = 10	**PQ**	**HQ**	**Problems**	**Satisfaction**
PQ	1.000			
HQ	.331	1.000		
Problems	-.712*	-.485	1.000	
Satisfaction	.801*	.621	-.634*	1.000

*p < 0.05

3.3.3. Analysis by Interaction Sequencing

To verify the influence of interaction sequencing on UX dimensions, we analyzed the results according to the group (inspection or testing) and script that each participant was assigned to. A previous study from Cockburn et al. (2017) indicated a significant influence of interaction sequencing with the recency effect on users' satisfaction. Users tended to be more satisfied in the *+end condition* than the *-end* one. Considering the small sample size for the inspection group for each condition (6 for +end and 5 for -end), we only performed statistical analysis for the testing group. In this sense, we derived the following hypotheses:

H$_5$: There is no difference in the satisfaction between the participants in *-end* and *+end conditions* for the testing group.

H$_{A5}$: There is a significant difference in satisfaction between the participants in the *-end* and *+end conditions* for the testing group.

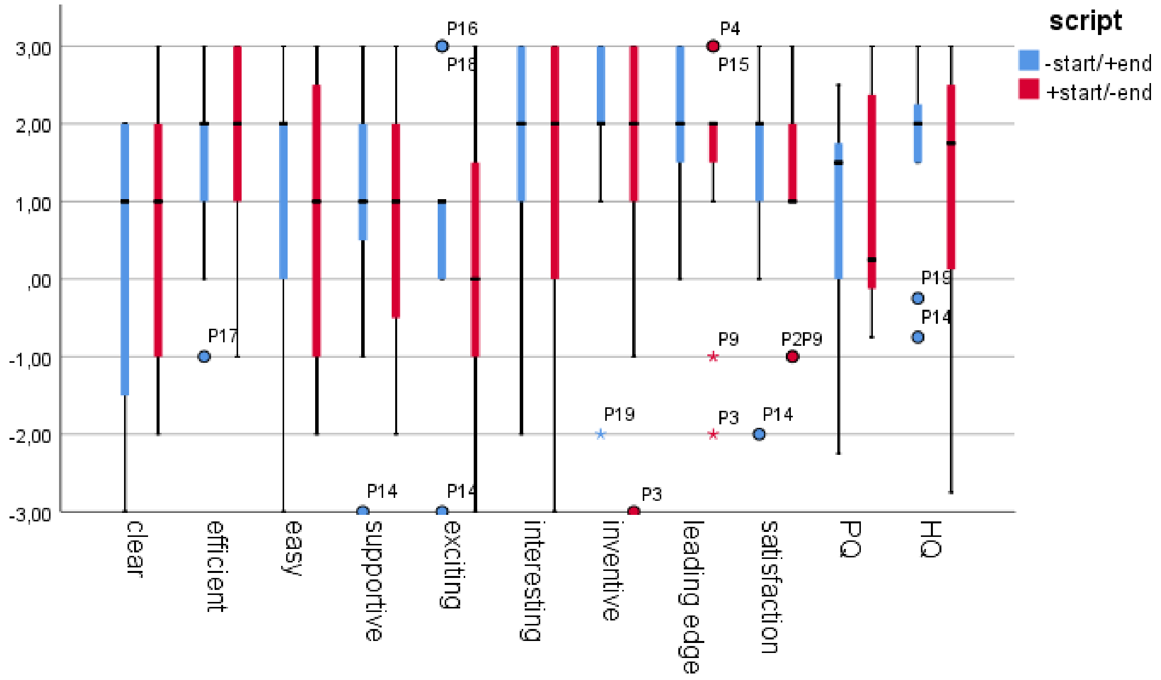


Figure 3.5. UX evaluation from the testing group by script.

Regarding the **testing group**, the results were very similar, with little differences between the *-end* and *+end conditions* (see Figure 3.5). Participants who followed the script with a negative end (*-end*) provided slightly lower ratings, mainly on HQ dimension. The statistical analysis, however, did not indicate significant difference between the two conditions, neither for the PQ dimension (U = 58, z = -.165, p = .869) nor for the HQ dimension (U = 57, z = -.231, p = .817), and overall satisfaction (U = 48.5, z = -.831, p = .406), thus retaining the H$_5$ null hypothesis.

Regarding the **inspection group**, the differences between the two conditions were a little more expressive (see Figure 3.6). Inspectors who followed the script with a negative ending (*-end*) tended to give lower ratings in all items evaluated. It may be due to the level of effort required from the inspectors in the *-end condition*. The long task at the end with many steps to perform, inspect, and identify problems may have influenced their perception of the application, leading to a more negative evaluation due to the recency effect. However, the small sample size impairs drawing conclusions.

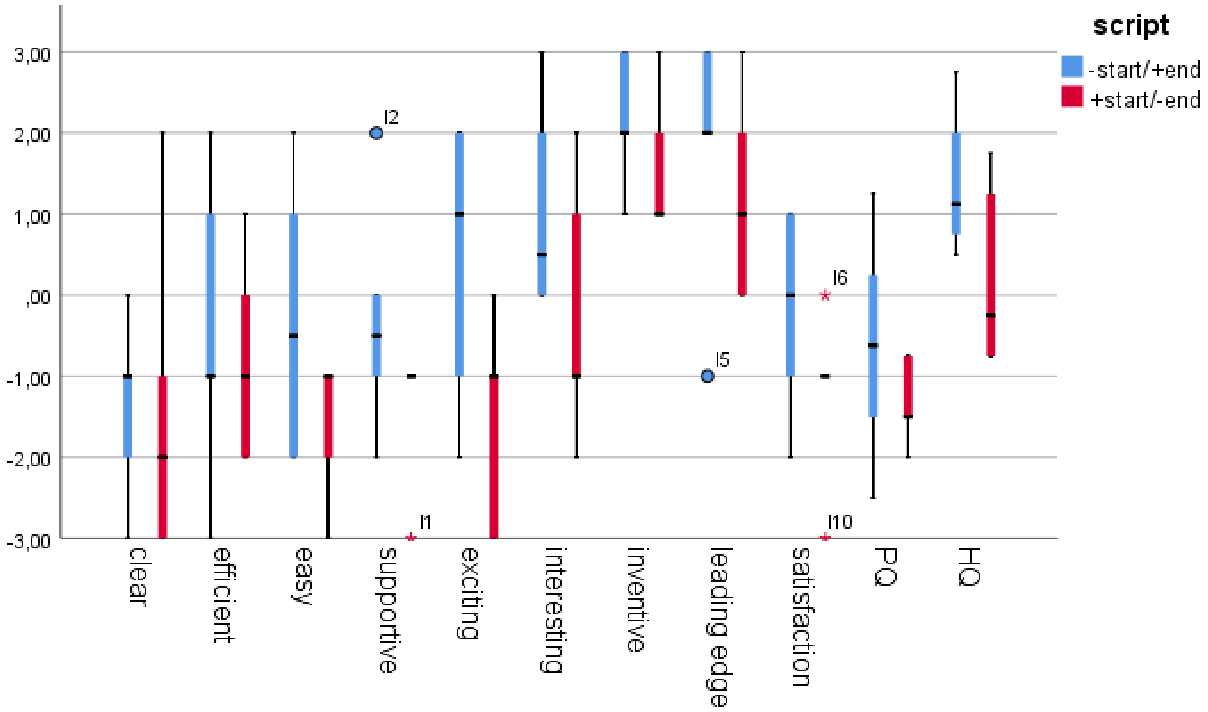


Figure 3.6 - UX evaluation from the inspection group by script.

3.3.4. Analysis by Previous Experience

Regarding the analysis considering the experience with similar shopping applications, we only analyzed the users' data from the **testing group**. It is because only one participant did not have previous experience with this type of application in the inspection group, not making it possible to perform a balanced comparison among the participants from this group.

According to Hassenzahl (Hassenzahl, 2004), the number of usability problems can affect the PQ dimension while HQ remains stable. In this sense, we expect participants with previous experience with similar products to face fewer usability problems, thus positively evaluating the PQ. By contrast, participants without prior experience would have difficulty interacting with the application, leading to a more negative PQ evaluation. Regarding the HQ dimension, we pose that both will evaluate the application positively due to its innovative approach. Finally, as satisfaction is the overall perception of the product, we expect participants with previous experience to be more satisfied than participants without prior experience. Thus, we derived the following hypotheses:

H_6: There is no difference in the PQ dimension between participants with and without previous experience with similar applications.

H$_{A6}$: There is a significant difference in the PQ dimension between participants with and without previous experience with similar applications.

H$_{7}$: There is no difference in the HQ dimension between participants with and without previous experience with similar applications.

H$_{A7}$: There is a significant difference in the HQ dimension between participants with and without previous experience with similar applications.

H$_{8}$: There is no difference in satisfaction between participants with and without previous experience with similar applications.

H$_{A8}$: There is a significant difference in satisfaction between participants with and without previous experience with similar applications.

The analysis revealed that participants without previous experience evaluated the application positively in both PQ and HQ dimensions (see Figure 3.7). Those who had already used similar applications before also reported a positive experience on the items evaluated by the HQ dimension, with quite similar ratings. By contrast, they tended to have a neutral perception of the PQ dimension items.

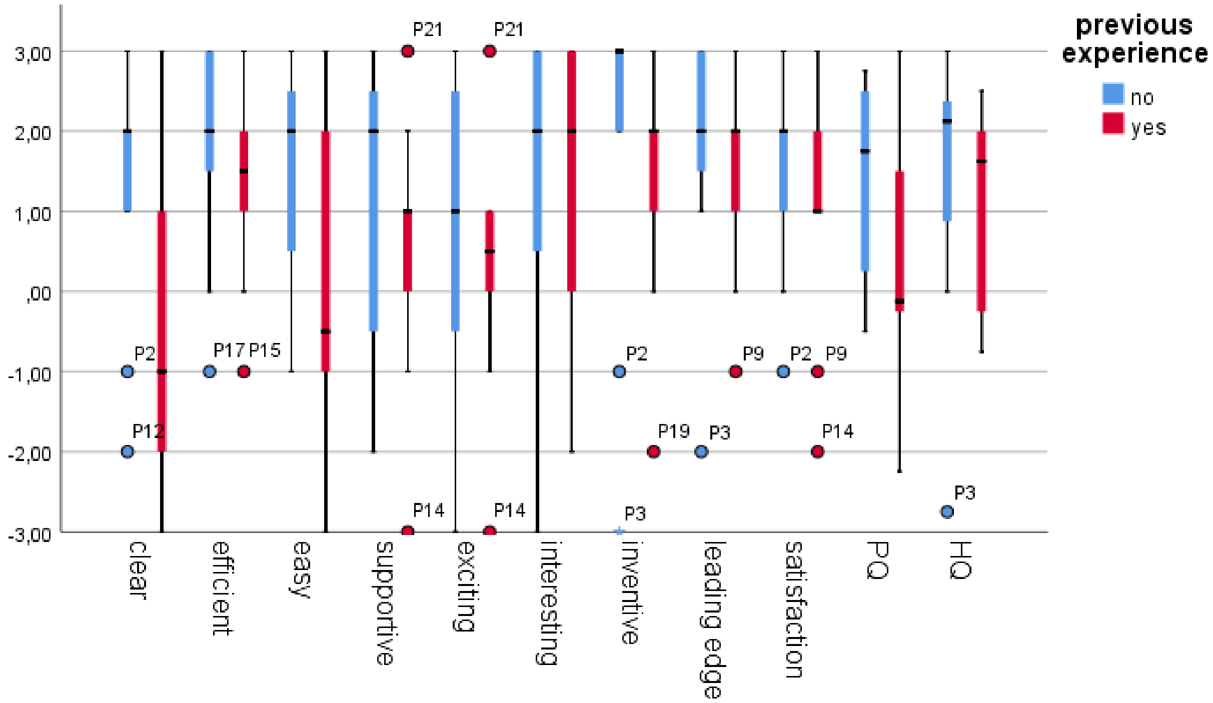


Figure 3.7 - UX evaluation from the testing group by previous experience with similar applications.

The statistical analysis revealed a significant difference in the PQ dimension between the two groups with a medium effect size (*U* = 30.5, *z* = -1.956, *p* = .050, *r* = .417). This indicates that opposing our hypotheses, the participants with previous experience gave lower ratings for the PQ dimension than those without previous experience, thus rejecting the H$_{6}$

hypothesis. However, no significant differences were found neither for HQ dimension (*U*= 43.5, *z* = -1.094, *p* = .274) nor for overall satisfaction (*U* = 42, *z* = -1.251, *p* = .211), thus retaining both H_7 and H_8 hypothesis.

To better understand the results, we analyzed each of the problems reported by the moderator in the user testing session and abstracted them to organize into groups of similar problems by analyzing the context of the problem and keywords to assign codes. Consider, for instance, the following problem descriptions: "*It is* ***unclear*** *where to* ***create the account****. Would it be on the 'continue' button?*" and "*User* ***did not understand*** *how to* ***complete the purchase****. He went and came back to the address screen a few times*". Both descriptions and the highlighted keywords indicate that these actions are not intuitive for the user. Thus, we assigned them the "Not intuitive action" code.

We identified that many of these problems were shared between participants from both groups. The four problems with the highest number of occurrences were as follows (see Figure 3.8): i) **not intuitive action:** when the participant does not know how to perform a given task, exploring the interface and activating different links in the app. Nine participants without experience and six participants with previous experience faced this problem; ii) **Difficulty in finding information (8 without experience / 6 with experience):** when the user knows what s/he is looking for but did not find it with ease or quickly. Eight participants without experience and six with experience had this difficulty; iii) **not visible feedback:** when the app provides feedback to the user but s/he cannot see it because it is not visible (e.g., when the user clicks in a given option, the chat screen does not scroll down and the loaded information does not show up); and iv) **not clickable element:** when the user tries to click on an element on the screen but it does not have any functionality (e.g., there is an icon of a GPS in the address information form, but it does not have any functionality). Five participants from both groups faced this issue.

We also compared the results between the ratings provided for the application being evaluated and for a similar application that the participants used before (see Figure 3.9). The results indicated that, in general, their remembered experience with a similar app was positive, mainly on the PQ dimension. By contrast, the participants found that the application being evaluated was not so clear and not so easy. However, it is interesting to mention that the participants found the evaluated app more inventive and leading-edge than the apps they had used before, highlighting that users recognized its innovative approach to using a chatbot.

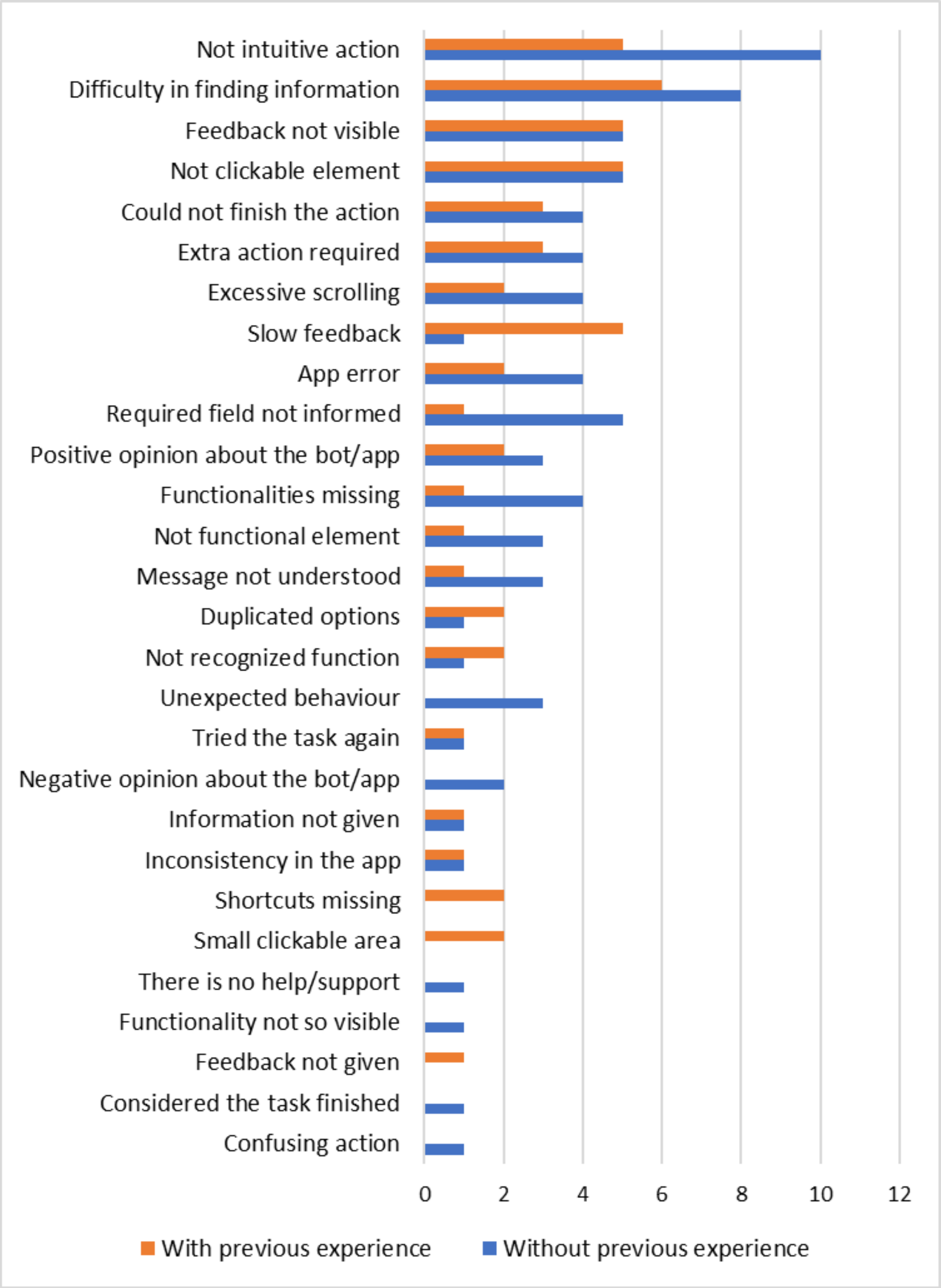


Figure 3.8 - Main problems faced by users from both groups.

To better understand how each UX dimension are correlated to users' overall satisfaction according to previous experience, we conducted a correlation analysis. Table 3.11 presents the correlations for the participants from user testing who already used similar apps. The results indicated that overall satisfaction is strongly and moderately correlated with both PQ (r = .962, p < .0001) and HQ dimension (r = .709, p = .015) respectively. By contrast, the correlation between HQ and overall satisfaction was strong (r = .833, p < .001) for the group without previous experience (Table 3.12). However, no correlation with PQ dimension was found (*r* = .546, *p* = .082). There was also no correlation between the number of problems and PQ for none of the groups.

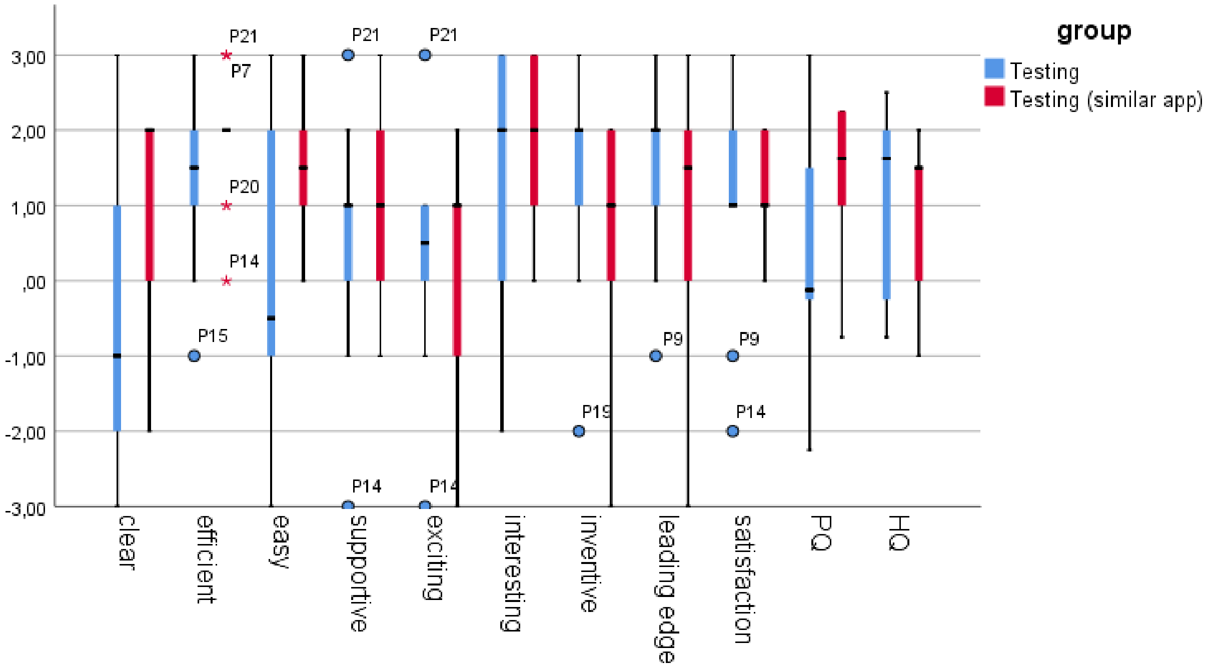


Figure 3.9. Comparison between the ratings for the evaluated app and the similar app.

Table 3.11. Correlations for the testing group who already used similar applications.

n = 11	PQ	HQ	Problems	Satisfaction
PQ	1.000			
HQ	.593	1.000		
Problems	-.492	-.58	1.000	
Satisfaction	.962**	.709*	-.531	1.000

** p < 0.01 *p < 0.05

Table 3.12. Correlations for the testing group without previous experience.

n = 11	PQ	HQ	Problems	Satisfaction
PQ	1.000			
HQ	.315	1.000		
Problems	-.287	.133	1.000	
Satisfaction	.546	.833**	.163	1.000

** p < 0.01

These results indicate that overall satisfaction of users who never used similar applications before is strongly associated with hedonic aspects. The more innovative and interesting the application is, the greater is users' satisfaction with it. In turn, for users who already had previous experience with similar applications, both pragmatic and hedonic aspects play an important role on their satisfaction, with stronger emphasis on the former. The lack of a significant correlation between the number of problems and PQ may indicate that not all problems annotated by the moderators may be considered an actual problem or may have low severity from the users' point of view.

3.4. DISCUSSION

We divided the discussion of the results into two main subsections. First, we discuss the results from the two main types of evaluation methods: inspection (UX-Tips) and testing (CTA). In the next subsection, we walk through each of the analyzes we carried out from the UX perspective.

3.4.1. Inspection vs Testing

The comparison between the two approaches demonstrated that inspection identifies a greater number of problems than testing, being more effective. The UX-Tips method (inspection) also identified more unique problems and around 42% of the problems from CTA (testing), with half the number of participants. These results are consistent with previous studies (Law and Hvannberg, 2002; Maguire and Isherwood, 2018; Tan et al., 2009), in which inspections using Heuristic Evaluations addressed more problems than user testing. In this sense, inspection seems to still be the more cost-effective way to identify problems in interactive software products, making it a good choice for companies. By contrast, inspectors did not identify 45.1% of the problems, which were addressed only by user testing. Such a result indicates that these methods are complementary and using both will allow covering a wider variety of problems.

It is also worth mentioning that UX-Tips allowed identifying the sentiments associated to the problems faced by inspectors when using the application. This information may be useful for developers as they can identify not only the problem the user faced but also what type of sentiment and experience the product is providing. The same results were not achieved by usability testing with CTA, where only a few participants verbalized their thoughts about their experiences, although being encouraged to do so.

3.4.2. UX Evaluation

3.4.2.1. Evaluation Method Nature vs UX Evaluation

The results showed that inspectors tended to provide significantly lower ratings for the application in comparison to users, indicating the influence of the method on UX evaluation. As the goal of the inspection is to find as many problems as possible, the number of issues they found indirectly influenced their perception of the application, mostly on the pragmatic aspects, such as its ease of use and efficiency. We found similar results in an empirical study we conducted during the master's degree to evaluate a Web platform for a state traffic department (Nakamura et al., 2020). The main difference is that in that study, inspectors provided significantly lower ratings to all UX dimensions in comparison to users, not only for the PQ dimension, probably because the evaluated platform did not provide innovative or exciting features. Although identifying the greatest number of problems is desirable to improve the product's quality, this bias toward lower ratings reduces the reliability of UX evaluations from inspectors' perspectives. It is possible that many of the problems they found go unnoticed by users during actual interaction, thus not being a faithful representation of users' experience.

Regarding HQ dimension, the ratings were more similar. Participants from both groups perceived it as innovative, even those who already used other shopping applications. Only two inspectors reported violations in the Innovation item, which definition is: "The application has innovative features (different ways to meet the user's need)". Thus, the majority of the inspectors agreed that the application was innovative. The smaller variability in the HQ results from both groups, mainly in the 'leading edge' and 'inventive' adjectives, also reinforces this finding.

3.4.2.2. Interaction Sequencing

The results from the analysis by considering the interaction sequencing revealed no significant differences between the *-end* and *+end conditions*, neither for the inspection group, nor for the testing group. However, although statistical differences were not found, the participants who followed the negative end script tended to provide lower ratings in both groups, mainly on the inspection one. This partially supports the findings from Cockburn et al. (2017) regarding the recency effect, in which the last episode weights more in overall people's remembered experience. As the last task from the negative ending script lasts longer due to its greater number of actions to be performed, it may have influenced the perception of inspectors who followed this script, given that, at each action, s/he needs to inspect the whole interface.

In the testing group, this difference was subtle. Given that the interaction during user testing is fluid, without the interruption to look for problems, the sequencing effect may have had a lower impact on users' perception. This may indicate that, during everyday use of an interactive software product or during UX evaluations with software applications already developed and available in the market, the sequencing effect may not have a strong impact on users' experience as expected.

3.4.2.3. Previous Experience

Participants with previous experience evaluated PQ items with significantly lower ratings than those who never used similar applications before. This finding opposes with the results obtained by previous works such as from Kim et al. (2015) and Sagnier et al. (2020), where users with previous experience tended to evaluate their experience significantly more positively than those with no experience. In our study, participants with prior experience gave lower ratings for the pragmatic attributes, while participants who never used similar apps provided higher ratings for this dimension.

Overall, the results revealed that the app is not so intuitive, as both participants with and without experience faced difficulties. Additionally, their previous experience might have intensified their negative perception of the app, resulting in contradictory evaluations between the two groups. The results presented earlier in Figure 3.9 indicated that the similar app was considered easy to use, which may have influenced their ratings, as they faced a considerable number of problems during their interaction. Regarding the group without previous experience, the novelty of interacting with a chatbot may have overcame the difficulties they faced during the interaction, especially because they did not have any prior experience baseline for comparison.

When analyzing the number of problems, both groups had very similar results. Participants without previous experience had, on average, 8.91 problems, while those with previous experience had 9.73. This reinforces that the previous experience acquired from using similar apps did not have much influence in their skills to perform tasks with the evaluated app, as the interaction between a conventional shopping app is very different. A recent example of changes in interaction that led to several criticisms was the new Start menu in Windows 8, as Schneidermeier et al. (2014) reported. The button was not accessible from the taskbar anymore and its interface switched to a tablet mode in full-screen, with big blocks representing different programs and functionalities. This change was not well received by the public and the company brought back the old Start menu style in Windows 8.1, which remained until recent versions of this operational system. In this line, a study from Martens and Johann (2017) who analyzed app reviews from Apple AppStore revealed that a complete redesign of an app can lead previously positive sentiment to turn into a negative sentiment, which can explain the low scores from experienced users in our study. As they were used to the traditional form of interaction when shopping through an app, the paradigm shift may have led them to have a more negative perception about the application in the pragmatic dimension due to the difficulties they faced with the new approach.

In general, our results suggest that Hedonic and Pragmatic aspects have different influence on users' satisfaction. Jang and Yi (2017) addressed this issue before by extracting UX aspects from online user reviews and performing a regression analysis with the ratings. Their results indicated that hedonic aspects had a positive effect on user satisfaction for all products evaluated. By contrast, pragmatic aspects had varied effects on user satisfaction. A major difference between their work with ours is that it is not possible, for example, to identify the influence of factors such as previous experience and interaction sequencing on user satisfaction through online user reviews, as their background and actual interaction are unknown. Our study revealed that HQ and PQ dimensions have different effects on users' satisfaction according to their previous experience with similar applications, which may explain the varied effects of PQ dimension on user satisfaction obtained by Jang and Yi (2017). The correlation analysis indicated that while novices' satisfaction is associated with HQ aspects, the satisfaction of users with previous experience is associated with both PQ and HQ aspects, with greater emphasis on the PQ aspects. Given that the latter already had used similar products, their expectations were probably high, making it harder to satisfy them. In this sense, they were more critical about the app, considering both PQ and HQ when evaluating their satisfaction.

3.4.3. Implications

We detail below the implications for both practitioners and researchers regarding each of the discussed sections.

3.4.3.1. Implications for Practitioners

Regarding the comparison between the two methods (UX-Tips and Concurrent Think-Aloud – CTA), UX-Tips identified a greater number of problems and allowed the mapping between negative emotions and problems through the items related to hedonic aspects, something that is not possible or common when inspecting with traditional usability evaluation methods. Besides, UX-Tips addressed even more emotions than CTA, making it a good alternative especially to companies with confidentially issues that impairs the conduction of the test with real users.

The influence of the evaluation method in the UX evaluation indicates that practitioners should not evaluate the UX from the point of view of inspectors after inspecting an application. The number of problems identified during inspection can bias their perception about the overall experience and lead to inaccurate results when employing quantitative UX evaluation methods. Instead, practitioners should evaluate UX from the point of view of users or apply inspection methods that also address UX aspects like UX-Tips.

Finally, the results revealed that shifting paradigms by changing how users interact in the application might lead to negative evaluations when the proposed approach is not so intuitive, mainly on those who have previous experience with similar applications. This issue is critical, especially to companies that desire to launch a new application in the market and attract users of competing apps. Although including innovative features is essential to have a differential towards the competitors, it should be analyzed with caution. When major changes occur, practitioners should engage in conversation with users or explain the changes within the application itself, for example, by using tutorials and tooltips to make them familiarize with the new release (Martens and Maalej, 2019).

3.4.3.2. Implications for Researchers

The results of the UX-Tips method revealed that it is possible to map usability problems and the emotions that it awakens. It also demonstrated that the method is cost/effective to identify usability and UX issues. Further research can be conducted to investigate whether the method can be employed by novice inspectors. By doing so, it will be possible to assess its cost/effectiveness better and whether it is suitable to companies with budget constraints.

Regarding the influence of the evaluation method on UX, researchers should be aware of this effect when carrying out UX evaluations. Our study demonstrated that inspectors' perception of the experience after inspecting an application is not faithful, as the number of problems identified and the mental effort necessary to perform this type of task can lead to a more negative perception about the app. Researchers should avoid performing such evaluations in their studies.

Regarding interaction sequencing, this study indicated that it may not have significant effect during actual user interaction with a real application. Further studies can be conducted to corroborate or question our findings by manipulating the interaction in different types of applications.

The differences between UX ratings from users with and without previous experience when shifting interaction paradigms raise new research opportunities. Researchers can investigate, for instance, what changes on user interface can lead to more positive or negative UX according to their previous experience. One can also conduct longitudinal studies to investigate whether the change in the perception of the experience due to the paradigm shift persists over time, even after the user gets familiar with the new form of interaction.

Overall, the influence of factors, such as the method employed and previous experience, highlights that just the concept of hedonic and pragmatic dimensions is not enough to explain why users keep evaluating their UX as positive even when facing many interaction problems. Different factors should be investigated to identify which of them affect users' perceptions of their experience and lead to more positive or negative evaluations. By doing so, it will be possible for developers to develop products that convey pleasurable experiences by focusing on factors that positively affect the UX while mitigating the effect of factors that deteriorate the UX.

3.5. SUMMARY

This chapter presented a UX evaluation study to investigate the influence of previous experience, interaction sequencing, and number of problems on users' perceptions about their experience. To do so, we employed three evaluation methods (inspection, testing, and a UX evaluation questionnaire) to assess the UX of a mobile software application designed to facilitate online shopping, developed by a local development company. Investigating the influence of such factors on users' subjective judgment is important, as practitioners make design decisions based on the feedbacks received from users. By knowing which factors can affect users' perceptions about their experience, practitioners can comprehend the results better, reducing the risk of misinterpreting them and making bad design decisions that can compromise future releases. Our results revealed that previous experience and the method employed can influence on the ratings of UX dimensions and overall satisfaction. These findings highlight the importance of considering these factors when conducting UX evaluations to better interpret and get reliable results. We present our main findings below and further research possibilities.

Our results provided evidence that **previous experience with similar applications** affect how users evaluate their UX, as they can use it as a baseline to evaluate their experience with the product under evaluation. Although there is an understanding that Pragmatic and Hedonic are two main dimensions that composes UX, it seems that each is influenced differently depending on the factor. The overall satisfaction of users who already used similar applications, for example, were more related to pragmatic aspects, while for users without experience, the satisfaction was associated with hedonic aspects. In this sense, if a company desires to attract users from other similar applications, developers should primarily focus on fulfilling their do-goals, i.e. the achievement of tasks and goals (Hassenzahl, 2018a), then moving on to the be-goals, i.e. the potential of the product to support pleasure in use and ownership (Hassenzahl et al., 2010). Finally, regarding UX evaluations, researchers and practitioners should be aware of this factor when recruiting subjects to conduct UX evaluations in order to better understand the results. As each user profile may provide different feedbacks about the product, their data should be analyzed and interpreted accordingly.

Regarding **interaction sequencing**, our results did not reveal a significant influence on UX evaluation. Although previous studies in laboratory settings with applications designed specifically for the study reported a significant difference on the results (Cockburn et al., 2015, 2017), the influence of interaction sequencing seems to not be as strong as expected when using in the context of a software already developed and available in the market.

Finally, the results strengthen the findings on the **influence of the evaluation method** on UX evaluations, which we identified in the study we conducted during the master’s degree (Nakamura et al., 2020). Inspectors tended to give lower ratings for the PQ dimension in comparison to users, with strong negative correlation between the number of problems identified and their ratings. Although it can be seen as important to improve the quality of the application based on these lower ratings, it may lead developers to focus their efforts trying to improve UX aspects that actual users may not be concerned about, implying in unnecessary effort and cost. This indicates that researchers and practitioners should be aware of such effects when using different types of evaluation methods, as the perceptions of evaluators may have been influenced by the method’s evaluation process and not reflect the perceptions from actual users.

CHAPTER 4 – SYSTEMATIC MAPPING OF STUDIES ANALYZING USER REVIEWS FROM APP STORES

This chapter presents our second iteration over the relevance cycle, where we carried out a systematic mapping study to investigate what factors are reported in the literature and their associated polarity. The results allowed us to obtain a set of factors that served as the basis for our method.

4.1. INTRODUCTION

In the first empirical study, we aimed to investigate why users keep evaluating their experience as positive even when facing many interaction problems and expressing negative emotions. Our hypothesis was that there are factors that influence on users' perceptions of their experiences, leading sometimes to contrasting results, such as those found by Bopp et al. (2016). The results of our first study supported our hypothesis, indicating that factors such as previous experience with similar applications and the number of problems identified can affect UX in the context of mobile apps. It also revealed that interaction sequencing does not have a significant effect on users' perception about their experience when evaluating this phenomenon in real applications. All these findings allowed us to understand the effect of these factors and highlighted the importance of considering their effects to design strategies to develop software applications that bring pleasurable experiences. In this sense, we moved forward on this research, aiming to identify other factors that can affect UX.

This chapter presents our second iteration over the relevance cycle, which allows assessing the novelty of our research and identifying potential gaps to be explored. To do so, we performed a systematic mapping study to identify publications that report factors associated to positive or negative evaluations. The knowledge obtained from this iteration also leads to a first iteration over the rigor cycle, which ensures the rigor of the research necessary to build the artifact grounded in a solid theoretical foundation, and contributes to build a body of knowledge on the topic. In the rigor cycle, experiences and expertise that define the state of the art in the application domain are addressed, as well as the existing artifacts and processes (Hevner and Chatterjee, 2010).

As our goal is to investigate these factors in different types of products, we focused on publications that analyzed user reviews from app stores. Being considered as the "voice of the users", these reviews contain useful information for practitioners, such as user requirements, bug reports, and user experiences with specific app features, which can be used to drive the development of the application and improve future releases (Pagano and Maalej, 2013). We describe details of this systematic mapping study in the next subsections.

4.2. REVIEW PROTOCOL

Before starting our systematic mapping study, we developed a review protocol. This protocol defines the procedures to perform the systematic literature review, being an important document for both the validity and the practical conduct of the review (Wohlin et al., 2012). We present details of this protocol in the next subsections.

4.2.1. Research Question

In this systematic mapping, we aimed to answer the following main research question: "What are the UX-related factors that influence users' evaluations in app store reviews and how they affect UX?". We also defined the following sub-questions to answer specific questions related to: i) *dataset source*: to identify the target population; ii) *extracted information*: to identify which data were obtained for analysis; iii) *analysis methods*: to understand how the data was analyzed; iv) *data categorization*: to identify whether and how the data was organized; v) *scope of the analysis*: to verify how comprehensive the study was in terms of analysis and apps sample; vi) *identified factors and their associated polarities*: to identify whether the factor affects UX positively or negatively; and vii) *factor influence analysis*: to investigate the extent the impact of the factor was analyzed. Table 4.1 presents each research sub-question.

Table 4.1 - Research sub-questions.

Sub-question	Description
SQ1	What was the source of the analyzed reviews?
SQ2	What is the information extracted from the sources?
SQ3	Which methods were used to analyze the data extracted?
SQ4	Was the information categorized? How?
SQ5	What is the dataset size and analysis scope of the extracted publications?
SQ6	What are the identified factors and their associated polarity?
SQ7	Was the influence of the factor on user rating or sentiment analyzed? How?

4.2.1.1. Research Scope

We carried out this systematic mapping on the IEEE Xplore, ACM, and Scopus. While Scopus is a meta-library that indexes publications from several well-known publishers (e.g., Springer, Elsevier, and Taylor & Francis), ACM and IEEE are two main digital libraries from the computer science field. We selected these databases as they are recommended by previous systematic literature reviews as the adequate and relevant ones to use (Dyba et al., 2007; Mendes et al., 2020; Petersen et al., 2015). Additionally, we performed a one-step backward snowballing process, which consists of following the references from each selected paper to identify other relevant ones (Wohlin et al., 2012).

4.2.1.2. Language

We selected only publications written in English, given that most of the international conferences and periodicals adopt it as the main language. Additionally, English is the dominant language for global communication, thus making it possible to replicate and/or extend this systematic mapping study by other researchers.

4.2.1.3. Search Terms

We first defined a set of control papers that the search engines should return to build our search string. To do so, we analyzed the papers from a systematic mapping study conducted by Genc-Nayebi and Abran (2017) that addressed studies on app stores opinion mining and selected those that presented factors associated with positive, negative, or neutral evaluations.

We followed the procedure described by Kitchenham and Charters (2007) to define the terms of the research. They suggest defining five parameters: population, intervention, comparison, outcome, and context. Given that our focus is not to compare interventions, we did not use the comparison parameter. Table 4.2 presents the set of terms for each parameter defined below:

- **Population:** user reviews from app stores;
- **Intervention:** methods/techniques employed to analyze user reviews and identifying influencing factors;
- **Comparison:** not applicable, as the goal is to identify the factors from the literature;
- **Outcomes:** the effect of these factors on UX;
- **Context:** within the domain of mobile app stores.

To build the search string, we used the boolean operator “OR” between the words with similar ideas for each parameter and the boolean operator “AND” to join the four parameters (see Table 4.3). We tested the string several times with different combinations of words to reduce the number of publications that are not related to the research topic while ensuring that the set of reference publications is returned. To define the control set, we analyzed the 24 publications from a systematic literature review related to our topic (Genc-Nayebi and Abran, 2017). We identified four publications (S02, S03, S04, and S05) that report factors influencing users' ratings and sentients, which comprised our control set.

Table 4.2 – Terms selected to compose the search string.

Population	**Intervention**	**Outcomes**	**Context**
review	mining	experience	mobile app
opinion	analysis	UX	mobile apps
comment	processing	usability	mobile application
rating	examining	sentiment	app store*
			appstore*
			app marketplace
			app market
			app markets
			application market*

Table 4.3 – Final search string.

*(review **OR** opinion **OR** comment **OR** rating) **AND** (mining **OR** analysis **OR** processing OR examining) **AND** (experience **OR** UX **OR** usability **OR** sentiment) **AND** ("mobile app" **OR** "mobile apps" **OR** "mobile application*" **OR** "app store*" **OR** appstore* **OR** "app marketplace*" **OR** "app market" **OR** "app markets" **OR** "application market*")*

4.2.1.4. Selection Criteria

We defined a set of inclusion and exclusion criteria to select publications that are related to our research, i.e., publications that present factors associated to positive, negative or neutral evaluations (see Table 4.4). To analyze the papers, we first needed to define the concept of UX to delimit the scope of the factors related to UX. We adopted the definition from Hassenzahl and Tractinsky (2006), which, according to Lallemand et al. (2015), was the most preferred definition among practitioners and researchers. According to it, UX is "a consequence of a user's internal state (predispositions, expectations, needs, motivation, mood, etc.) the characteristics of the designed system (e.g., complexity, purpose, usability, functionality, etc.) and the context (or the environment) within which the interaction occurs (e.g., organizational/social setting, the meaningfulness of the activity, the voluntariness of use, etc.)." In this sense, we addressed factors related to the user (e.g., expectations, emotions, sentiment, demographics) and the app itself (e.g., bugs, features, functionalities). Finally, considering that we aimed to assess the effect of these factors on UX, we only included publications that reported the effect of the factor on users' ratings and/or reviews' sentiment, given that these two pieces of information convey the experience the user had with the application (Rodrigues et al., 2017).

Table 4.4 - Inclusion and exclusion criteria applied in the systematic literature review.

#	Inclusion criteria
IC1	Publications that present UX-related factors associated to negative, positive or neutral reviews from app stores.
#	**Exclusion criteria**
EC1	Publications that do not present UX-related factors or do not associate them with negative, positive or neutral reviews from app stores.
EC2	Publications that are not available for reading or collecting data (publications that are accessible only through payment or are not provided by the search engine).
EC3	Publications that are not written in English.
EC4	Books, proceedings, websites, and grey literature.
EC5	Duplicated publications.

The selection process comprised two steps called filters. In the first filter, we read the title and the abstract of each publication to select those related to our research topic. First, we assessed whether the paper addressed user reviews from app stores. Then, we analyzed whether it considered UX, either by explicitly mentioning it or by using other related terms (e.g., emotions, usability, satisfaction, sentiment). Finally, we analyzed whether the paper discussed the impact or influence of variables (factors) on UX, ratings, or sentiment. It is noteworthy that some publications did not discuss the results in the abstract, which did not allow us to know whether they identified the impact or influence of factors on UX. Thus, we decided to include such papers in the first filter to thoroughly read in the second filter to avoid missing important publications. Then, in the second filter, we fully read the publications included in the first filter. In both steps, we applied the inclusion and exclusion criteria to filter the publications.

To avoid the single researcher bias, we carried out the systematic literature review by involving two researchers. Before performing the first and second steps, the researchers independently classified, according to the inclusion and exclusion criteria, a sample of 20 randomly selected publications. Then, we evaluated the level of agreement between the researchers by applying Cohen's Kappa (Cohen, 1960) to ensure that the criteria are well defined and understood. The result indicated an almost perfect agreement (k = 0.89) according to the interpretation of Landis and Koch (1977) (see Table 4.5).

4.2.2. Data Extraction Strategy

After selecting the publications, we started to extract the data. To do so, we created an extraction form (see APPENDIX D) and followed the strategy proposed by Fernandez et al. (2011), which consists of defining a set of possible responses. We defined initial responses and refined them in an iterative process through the screening of the control set, as "important trends and ways of categorizing papers may only become evident as individual papers are read" (B. A. Kitchenham et al., 2015). Regarding analysis methods (SQ3), for instance, we identified the use of descriptive statistics (S02, S03, S04, and S05), statistical tests (S04 and S05), topic modeling (S03), sentiment analysis (S03), and manual analysis (S04 and S05). This strategy allows to standardize the extraction process, ensuring that the same data extraction criteria will be used for each sub-question, thus facilitating the classification.

Table 4.5 – Strength of agreement associated with kappa statistics.

Kappa 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

For **SQ1 (dataset source)**, we defined the following responses: a) **Google Play Store:** the dataset was obtained from the Google Play Store; b) **Apple AppStore:** the dataset was obtained from the Apple AppStore; c) **Windows Phone Store:** the dataset was obtained from the Windows Phone Store; d) **Other:** the dataset was obtained from other sources.

For **SQ2 (extracted information),** we defined the following responses: a) **Rating:** the star rating given by users when posting their review; b) **User review:** the comments users made with their opinions, complaints, and suggestions for the app; c) **App information:** app's metadata, such as release date, description, category, version, price, release/update notes, etc.; d) **Other:** information not directly obtained from the app store, such as app's project lifespan, number of commits, number of warnings in the source code, etc.

For **SQ3 (analysis methods)**, we defined the following responses: a) **Topic modeling:** consists of discovering relationships between documents (as well as the terms that compose these documents) and topics, making it possible to organize textual documents according to the topics discovered (Durelli et al., 2018); b) **Sentiment analysis:** is a method to explore the sentiment conveyed by people in textual data, determining whether the polarity of the text is positive, negative or neutral (Masrury et al., 2019); c) **Descriptive statistics:** when the authors describe and graphically present an overview of the dataset being analyzed, for example, by employing techniques to measure the central tendency (median, mean, mode), dispersion (frequency, variance, standard deviation), and dependency (linear regression, Spearman, Kendall and Pearson correlations) (Wohlin et al., 2012); d) **Statistical tests:** when the authors employed statistical tests (parametric or non-parametric) to test hypotheses and verify whether it is possible to reject a certain null hypothesis based on a sample from some statistical distribution (Wohlin et al., 2012); e) **Manual analysis:** consists of performing a coding process, for example, by employing content analysis to extract topics manually; f) **Other:** when employing other types of analysis, tools, and frameworks, for example, performing static code analysis by using a tool to address potential issues in the source code.

For **SQ4 (data categorization)**, we defined the following responses: a) **Yes:** when the authors grouped the data into categories during the analysis; b) **No:** when there was no any type of grouping. For each publication, we extracted the categories they used.

For **SQ5 (scope of the analysis),** we defined the following responses: a) **Individual:** the analysis was performed separately for each application; b) **Group:** the analysis was performed for a given group or different groups of apps; c) **General:** the analysis was performed in the whole set of applications, without separating them into groups or analyzing them individually. We also gathered information regarding the dataset size (number of apps, reviews, and categories).

For **SQ6 (identified factors and their associated polarity)**, we defined the following responses: a) **Positive:** if the factors identified were associated with positive reviews and/or ratings; b) **Negative:** if the factors identified were associated with negative reviews and/or ratings; c) **Neutral:** if the factors identified were neither associated with negative nor to positive reviews and/or ratings. For each polarity, we reported the factors associated with it

Finally, for **SQ7 (factor influence analysis)**, we defined the following responses: a) **Yes:** if the publication analyzed and reported the influence of the factors on users' evaluation, for instance, by performing statistical tests, correlation analysis or frequency distribution analysis; b) **No:** if the publication just reported the polarity of the factor, without assessing its influence on users' evaluation. For each factor, we extracted the reasons behind positive and negative evaluations.

Given that the search engines could return secondary studies, we created a specific form to extract this type of publication, which can be found in APPENDIX E. In this form, we defined questions to address specific information inherent to secondary studies, such as the search string used, queried databases, and results.

4.3. RESULTS

4.3.1. Selected Publications

Figure 4.1 presents the publication selection process. The search string returned a set of 562 publications, in which 253 were from Scopus meta-library, 167 from IEEE, and 142 from ACM. Among them, 132 were duplicated, resulting in a total of 430 unique publications. In the first filter (i.e. reading the title and abstract), 341 publications did not meet the inclusion criteria and were excluded. The remaining 89 publications followed to the second filter to be entirely read and submitted to the same inclusion and exclusion criteria from the first filter. A total of 71 publications did not meet the inclusion criteria and were excluded, resulting in 18 publications accepted in the second filter. It is noteworthy that some of the excluded publications presented factors but did not associate them with ratings or sentiments, which resulted in their exclusion (EC1). During the conduction of this systematic mapping study, we had to deal with a tradeoff between precision (i.e., the proportion of relevant studies returned) and sensitivity (i.e., the proportion of retrieved studies that are relevant studies) (Zhang et al., 2011). We decided to adopt a broader string to increase sensitivity at the cost of some precision to avoid losing relevant papers during our search, which is usually more desired (Zhang et al., 2011). As a result, we had many publications returned, but a small number of publications were included.

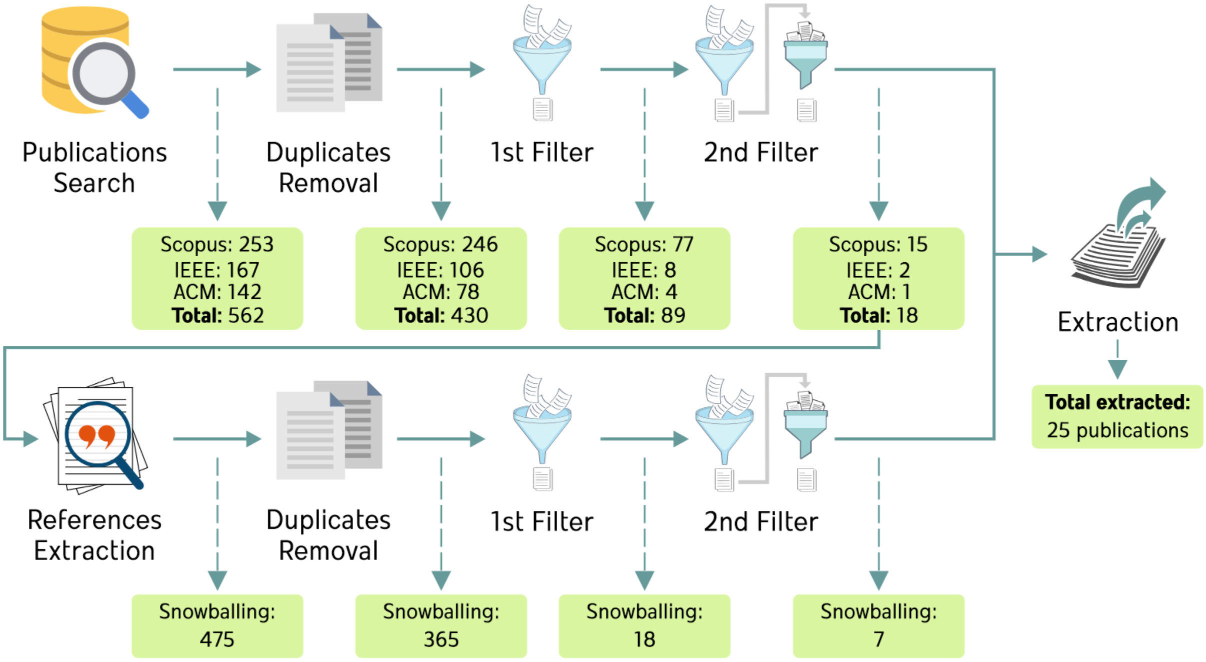


Figure 4.1 - Publication selection process.

Regarding the backward snowballing process, we extracted a total of 475 references from the 18 publications of the systematic mapping study. Among them, 110 were duplicated, resulting in 365 unique publications submitted to the same inclusion and exclusion criteria defined in the systematic mapping study. After applying the first and second filters, seven publications passed. At the end of the process, we extracted a total of 25 publications (databases + backward snowballing).

4.3.2. Publications Overview

The selected publications were published between 2012 and 2019. The graph presented in Figure 4.2 shows that the number of publications has grown since the first publication identified in 2012. The linear tendency, identified through simple linear regression, indicates a stable interest in investigating the reviews to understand better what are the positive, negative, and neutral aspects of the application. Given that we performed this systematic mapping study in April 2020, the data for this year is incomplete, which may explain its lack of publications.

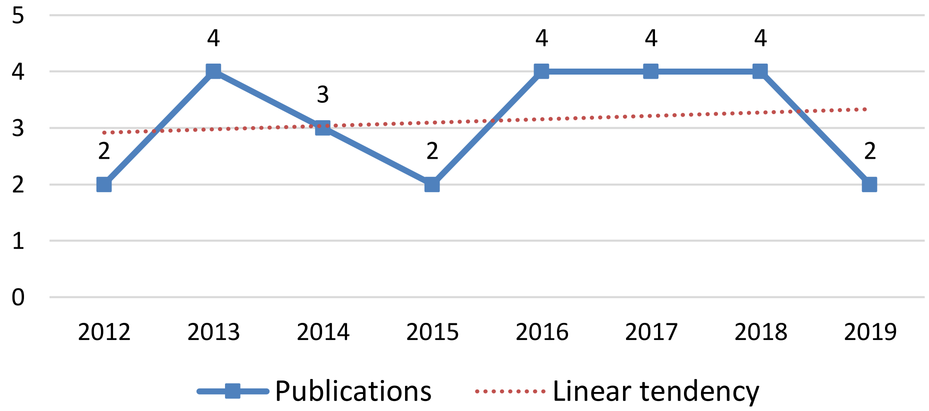


Figure 4.2 - Number of publications by year.

Most of the publications were published in conference proceedings (see Figure 4.3). Only five publications were published in journals and other three were presented in conference workshops. We also analyzed in which fields these works were published. To do so, we looked for the description, scope, and call for papers section of the venue's website and categorized them. We identified a total of 20 unique venues from nine different fields (see Figure 4.4). Most of the publications were in the Software Engineering field. The most active authors in the topic are Guzman, E., Hassan, A. E., Khalid, H., and Nagappan, M., with three publications each (see Table 4.6). The last three authors have also worked together in all three publications (S09, S10, and S13). The number of citations of the papers in this systematic mapping study (see Figure 4.5) reveals a considerable impact on the community. The publication from Guzman and Maalej (S06) has the greatest number of citations according to Google Scholar, with 576 citations, followed by the work from Pagano and Maalej (S05) and Khalid et al. (S10), with 550 and 447 citations each.

Table 4.6 - Ranking of the most active authors in the topic.

Ranking	Authors	Publications	Total
#1	Guzman, E.	S07, S23, S24	3
	Hassan, A. E.	S09, S10, S13	3
	Khalid, H.	S09, S10, S13	3
	Nagappan, M.	S09, S10, S13	3
	Luiz, W.	S20, S22	2
#2	Maalej, W.	S05, S07	2
	Palomba, F.	S19, S25	2
	Rocha, L.	S20, S22	2
	Shihab, E.	S09, S10	2

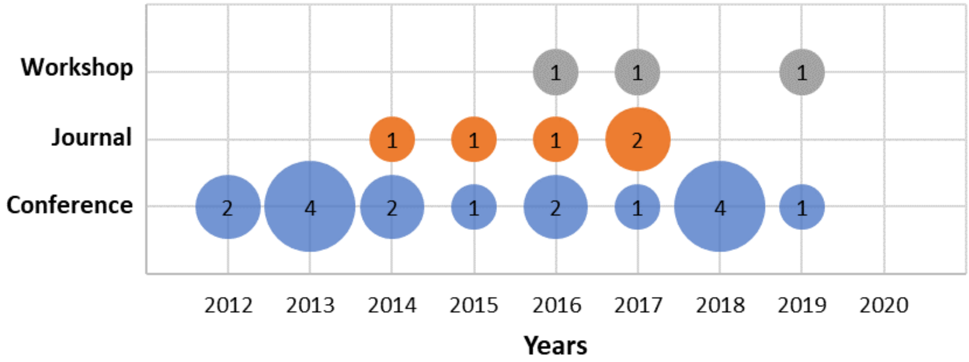


Figure 4.3 - Distribution of publications by venue and year.

In the next subsections, we present the results for each sub-question. It is worth mentioning that one of the publications (S17) was a secondary study. Thus, we extracted this publication using the extraction form for secondary studies and presented it separately in Section 4.3.10. Details of the mapping between the publications and each SQ can be found in APPENDIX E.

4.3.3. Dataset Source (SQ1)

Regarding dataset source, most of the reviews were obtained from Google Play Store, followed by Apple AppStore (see Figure 4.6). The reason to define the scope to a given app store is not informed in the publications we analyzed, except in the work of Pagano and Maalej [S05], where the authors explicitly mentioned that they explored the reviews from Apple AppStore due to their prior experience with the technology and its applications. The preference to explore reviews from Google Play Store may be related to the greater number of mobile devices with

Android operating system, holding a market share of around 72.6% against 26.7% of devices with iOS according to Statcounter GlobalStats[4] at May 2020.

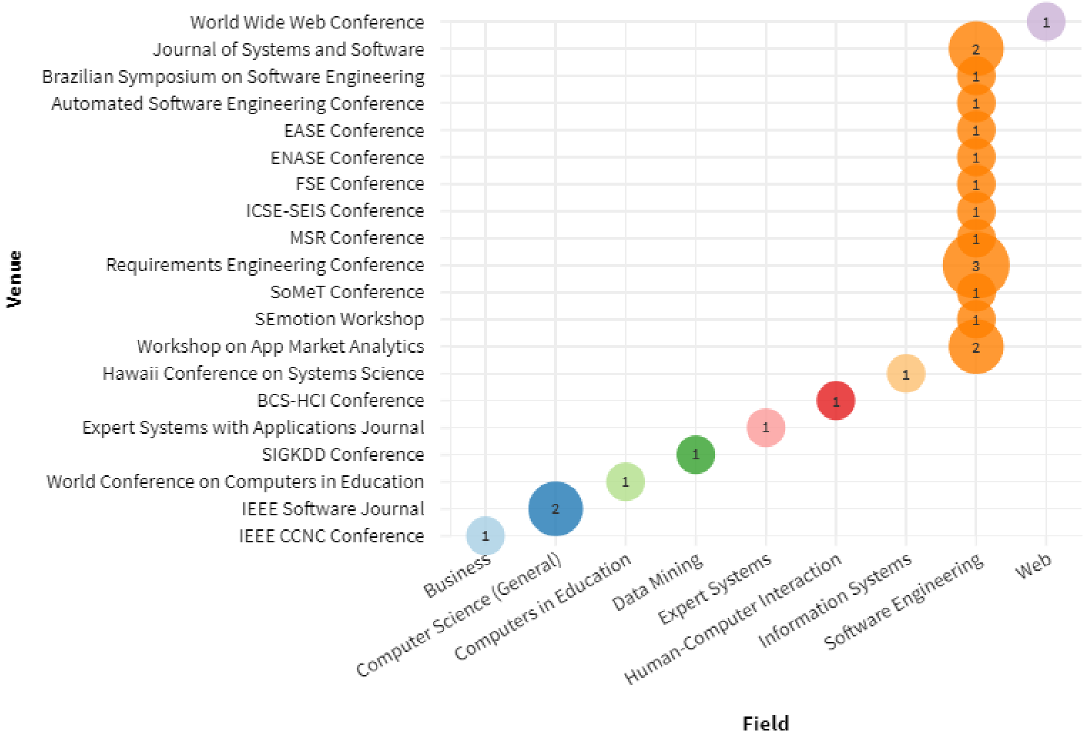


Figure 4.4 - Distribution of publications by venue and field.

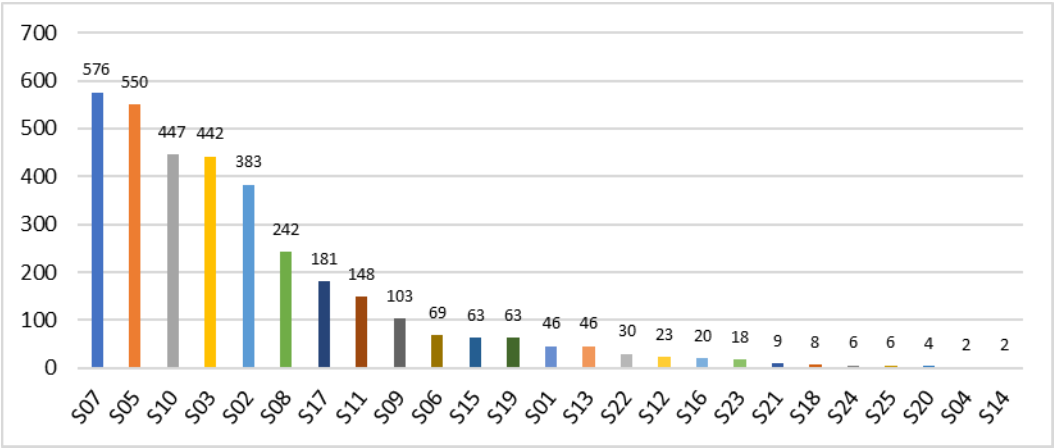


Figure 4.5 - Number of citations by publication.

The results also indicated a lack of studies involving different sources of information. Only Bano et al. [S07] and Guzman and Maalej [S18] analyzed reviews from both app stores. Two works used other data sources. Harman et al. [S02] extracted reviews from BlackBerry App World, while Kang and Park [S08] obtained the reviews from AppStoreHQ, a website that provides reviews of mobile apps from blogs, Twitter, and YouTube. Although the reviews of

[4] https://gs.statcounter.com/os-market-share/mobile/worldwide

the later are about iOS applications, we classified their dataset source as “other”, given that the reviews were not directly obtained from Apple AppStore. We did not find any work that analyzed reviews from Windows Phone Store. It may be due to its low popularity among mobile users and the small number of applications available for this operating system in comparison to Google Play Store and Apple AppStore. Additionally, the Microsoft, responsible to maintain the Windows Phone and Windows 10 Mobile OS discontinued them in 2017.

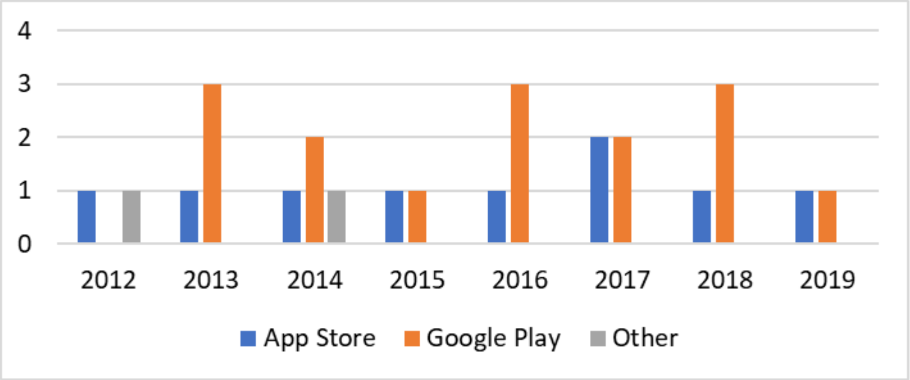


Figure 4.6 - Nnmber of publications by dataset source.

4.3.4. Extracted Information (SQ2)

On one hand, almost all publications extracted user reviews from app stores (see Figure 4.7). Only one publication (Harman et al. [S02]) did not obtain user reviews, but app descriptions, from where they extracted features to correlate them with other variables, such as price, app rating, and number of downloads. On the other hand, not all publications obtained user ratings for the analysis process. A possible explanation for this is because although star ratings provide a quick, direct, and objective overall evaluation of a particular app, it does not provide further details about the reasons why the app has, for instance, an overall score of three stars (Luiz et al., 2018).

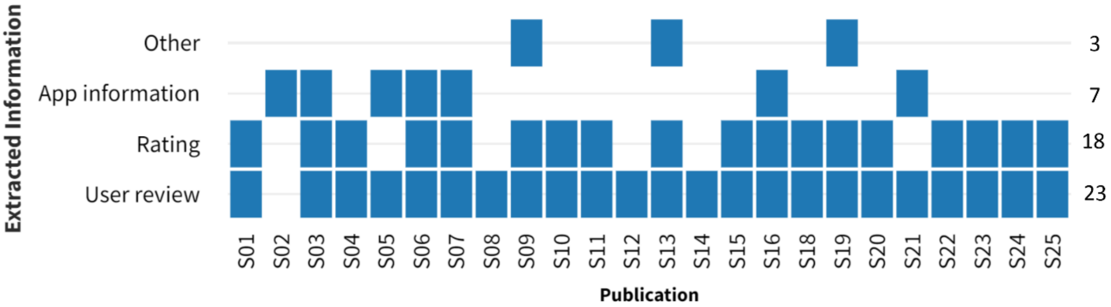


Figure 4.7 - Number of publications by extracted information.

Among the works that extracted both star ratings and user reviews, most of them (16 out of 18) performed some analysis involving star ratings and other variables. These analyses were carried out to identify inconsistent reviews (Fu et al. [S03], Luiz et al. [S22]), variations between ratings and sentiment analysis (Martens and Johann [S16], Luiz et al. [S22]), impact of different categories (Khalid et al. [S10]), type of app (healthcare vs non-healthcare) (Nicolai et al. [S25]), and features (Keertipati et al. [S15], Palomba et al. [S19]) on user ratings; as well as investigating correlations with gender (Guzman and Paredes-Rojas [S24]), culture (Guzman et al. [S23]), app version (Goul et al. [S01]), extracted topics from topic analysis (Ha and Wagner [S04], Iacob et al. [S06], Vu et al. [S11]), use of test cases (Durelli et al. [S20]), device model (Khalid et al. [S09]), and source code warnings (Khalid et al. [S13]).

Seven publications obtained app information such as app update dates, version, price, release notes, description, etc. Finally, three publications obtained other information and data, such as the Android Package Kit (APK) file to examine the code by using the FindBugs tool (Khalid et al. [S13]), app changelogs from Jira and Bugzilla (Palomba et al. [S19]), and users' device model (Khalid et al. [S09]).

4.3.5. Analysis Methods (SQ3)

Descriptive statistics was the most used method to analyze the data (see Figure 4.8). Five works (Goul et al. [S01], Harman et al. [S02], Khalid et al. [S09], Martens and Johann [S16], Guzman et al. [S23) performed correlation analysis to identify relationships between different variables, such as prices, downloads, culture, and ratings. Four works (Ha and Wagner [S04], Pagano and Maalej [S05], Iacob et al. [S06], Durelli et al. [S20]) identified the frequency of different variables to identify their distribution, such as the number of reviews by star ratings and factor. Two works employed regression analysis, one of them to detect inconsistent reviews (Fu et al. [S03]) and the other to identify features to prioritize (Keertipati et al. [S15]).

Sentiment analysis was the second most employed method to analyze the data and identify the sentiment of the reviews. The most used technique to analyze the sentiment of user reviews was SentiStrength, employed in 3 studies (Guzman and Maalej [S07], Shah et al. [S12], Martens and Johann [S16]). Each of the remaining works employed different sentiment analysis techniques and tools, such as Stanford CoreNLP (Nicolai et al. [S25), self-developed sentiment analysis (Durelli et al. [S20], Li et al. [S21], Fu et al. [S03]), Appbot (Bano et al. [S18]), RapidMiner (Mohan et al. [S14]), Clarabridge's tool suite (Goul et al. [S01]), SACI tool (Luiz et al. [S22]).

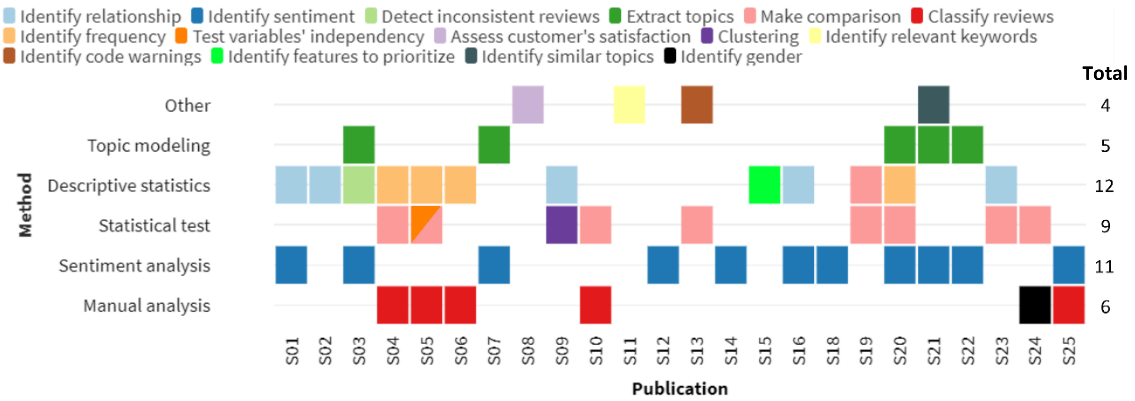


Figure 4.8 - Number of publications by analysis method.

We identified nine publications that performed some type of **statistical test**. Most of them employed Mann-Whitney, Wilcoxon Rank-sum or Chi-square to make comparisons between two groups, such as positive and negative factors (Ha and Wagner [S04]), review length and price (Pagano and Maalej [S05]), high vs low rated apps (Khalid et al. [S10], Khalid et al. [S13]), apps with great vs small number of implemented suggestions from reviews (Palomba et al. [S19]), apps with vs without test cases (Durelli et al. [S20]), and gender (Guzman and Paredes-Rojas [S24]). One work tested between multiple groups by employing Kruskal-Wallis, Chi-square, and Tukey-Kramer (Guzman et al. [S23]). The authors aimed to investigate whether there is a difference between distinct countries and the following variables: sentiments, rating, review content, review length, post time delay since release, gender, and factors (bug report, feature request, and 'other'). There was also one publication (Pagano and Maalej [S05]) that employed Chi-square to test variables' independency (e.g., a given factor vs ratings) and other that applied Scott-Knott test for clustering purposes (Khalid et al. [S09]), i.e., to identify groups of devices that are more prone to obtain lower ratings.

The **manual analysis** placed fourth. Five out of six publications (Ha and Wagner [S04], Pagano and Maalej [S05], Iacob et al. [S06], Khalid et al. [S10], Nicolai et al. [S25]) performed a manual coding process to tag the sentences and identify categories to classify the reviews. One publication performed a manual analysis to identify users' gender (S24).

Five publications employed **topic analysis** to extract topics (features) through unsupervised learning. Three works employed the Latent Dirichlet Allocation (LDA) unsupervised model (Fu et al. [S03], Guzman and Maalej [S07], Li et al. [S21]), and two works (Durelli et al. [S20], Luiz et al. [S22]) employed Non-negative Matrix Factorization (NMF) with Semantic Topic Combination (SToC).

Finally, four publications performed **other types of analysis** for a variety of purposes. In one publication (Kang and Park [S08]), the authors proposed an approach called VIKOR that employs sentiment analysis to assess customer satisfaction. In another (Vu et al. [S11]), the authors applied Vector Space Model (VSM) and K-means to cluster and identify relevant keywords. One publication (Khalid et al. [S13]) identified code warnings from the app source-code using the FindBugs tool to identify correlations between different warning categories and low ratings. Finally, one publication (Li et al. [S21]) calculated the Jaccard similarity index to identify similar topics.

4.3.6. Data Categorization (SQ4)

Half of the publications performed some type of categorization in the data they analyzed rather than presenting specific features of the application. However, we did not find any standardization. Only two studies categorized the reviews according to categories defined in previous work. Martens and Johann [S16] selected a sample of four categories from Pagano and Maalej [S05]: Bug Report, Feature Request, User Experience, and Rating. In turn, Nicolai et al. [S25] employed all the six categories defined by Panichella et al. (2015): Feature Request, Opinion Asking, Problem Discovery, Solution Proposal, Information Seeking, and Information Giving. They also included four new categories: Complaints, Compliments, Problem Reporting, and Noise.

Most of the categories defined were related to features/functionalities, problems/bugs, and users' positive/negative perceptions about the app. Eight publications defined categories related to features and specific functionalities, such as "feature request," "feature removal," "update," and "search" (Fu et al. [S03], Ha and Wagner [S04], Pagano and Maalej [S05], Kang and Park [S08], Khalid et al. [S10], Vu et al. [S11], Bano et al. [S18], Nicolai et al. [S25]). Other eight publications defined categories related to problems and bugs, such as "bug report," "problem reporting," and "functional error" (Ha and Wagner [S04], Pagano and Maalej [S05], Khalid et al. [S10], Vu et al. [S11], Khalid et al. [S13], Martens and Johann [S16], Guzman et al. [S23], Nicolai et al. [S25]). Seven publications defined categories related to UX aspects, such as user experience (Pagano and Maalej [S05], Martens and Johann [S16]), usability (Iacob et al. [S06]), performance (Khalid et al. [S13]), attractiveness (Fu et al. [S03]), adjective (Ha and Wagner [S04]), and complaints/compliments (Nicolai et al. [S25]). However, none of these publications analyzed the reviews through the lens of UX theory by considering pragmatic aspects related to the user's efficient and effective task achievement and hedonic aspects related to the user's emotions and sentiments. Although Pagano and Maalej [S05] and Martens and Johann [S16] defined a category named user experience, it is not related to the users' feelings and emotions as defined by UX theory, but to descriptions of the app in action, i.e., use cases where the application has proven to be useful.

4.3.7. Scope of the Analysis (SQ5)

In this subsection, we present the results regarding analysis focus (individual, category, and general) and the dataset size (number of apps, reviews, and categories)..

Half of the publications derived general conclusions by analyzing apps from different categories (see Figure 4.9). For instance, Guzman and Paredes Rojas [S24] analyzed user reviews coming from 7 countries that speak English. They obtained the reviews from 7 apps of different categories in Apple AppStore and investigated whether gender influences user's rating, time to post a comment after an app release, and review's length, sentiment, and content. In another work, Pagano and Maalej [S05] gathered a total of 1,126,453 reviews from 25 free and paid apps from 22 categories of the Apple AppStore. They investigated which of the 14 categories they defined are associated to more positive or negative reviews.

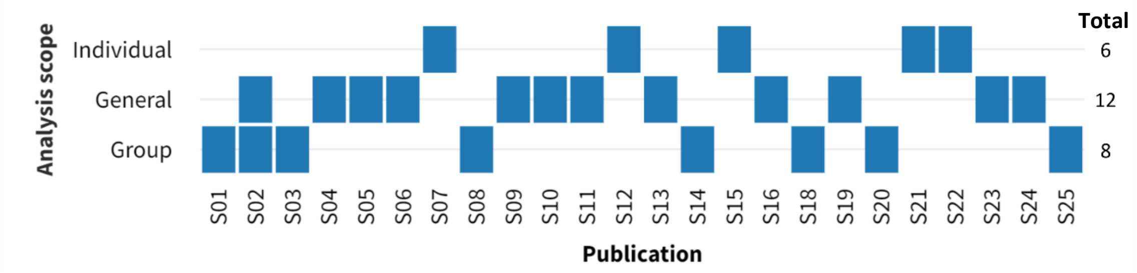


Figure 4.9 - Number of publications by data analysis organization.

Eight publications analyzed the reviews by groups. Some authors performed comparisons between different categories of apps, such as healthcare vs. non-healthcare apps (Nicolai et al. [S25]), with test cases vs. without test cases (Durelli et al. [S20]). Other authors focused on analyzing specific app categories such as educational applications (Bano et al. [S18]), mobile banking (Mohan et al. [S14]), social networking (Kang and Park [S08]), and productivity (Goul et al. [S01]). There was only one study that investigated a variety of categories of apps. Fu et al. [S03] analyzed 171,000 apps of 30 categories from app stores. They performed topic modeling and obtained the top-10 causes (topics) for negative reviews. For each app category, the authors identified the three topics that users complained the most.

Finally, six publications performed the analysis at the individual app level. For instance, Li et al. [S21] analyzed 1,148,032 reviews of WhatsApp from the Google Play Store. They aimed to investigate the correlation between users' positive and negative reviews before and after a sequence of apps' releases. To do so, they performed topic analysis to identify similar topics between different reviews and performed sentiment analysis over time.

Regarding dataset size, it varied considerably across the studies (see Table 7). Regarding the app sample, the analysis varied from a single app (S15, S21) to more than 170,000 apps (S03). This discrepancy is even greater when considering the number of reviews. Some publications analyzed less than a thousand reviews (S02, S04), while others analyzed millions (S03, S05, S11, S13, S16, S21).

The work of Fu et al. (S03) had the largest dataset with more than 13 million reviews from 171 apps. They performed topic modeling to group related words, and sentiment analysis to investigate their impact on ratings and identify inconsistent reviews. Through topic modeling, the authors identified the top 3 reasons for negative reviews in each of the 30 categories from Google Play Store. The large dataset comprising a variety of apps from different categories strengthens the findings of the study.

Other works, in turn, had small sample sizes. Ha and Wagner (S04), for instance, analyzed 556 reviews from 59 apps. They performed manual content analysis to classify the reviews into categories and performed statistical tests to investigate their impact on ratings. Although the results of some tests were statistically significant, the small sample size reduces the statistical power (Wohlin et al., 2012), especially because their analysis scope was general, that is, across all categories and apps.

Table 4.7 - Dataset analysis of the publications returned in this systematic mapping study.

Paper ID	Source	Apps	Categories	Reviews	Analysis scope
S01	Apple AppStore	9	1	5,036	Group
S02	BlackBerry App World	32,108	19	-	General Group
S03	Google Play Store	171	30	13,000,000	Group
S04	Google Play Store	59	30	556	General
S05	Apple AppStore	1,100	22	1,126,453	General
S06	Google Play	161	6	3,279	General
S07	Google Play Store Apple AppStore	7	Unspecified	32,210	Individual
S08	AppStoreHQ	8	1	1,487	Group
S09	Google Play	99	4	206,751	General
S10	Apple AppStore	20	15	6,390	General
S11	Google Play	95	Unspecified	2,106,605	General

Paper ID	Source	Apps	Categories	Reviews	Analysis scope
S12	Apple AppStore	25	Unspecified	100,000	Individual
S13	Google Play Store	5	All Google Play categories	2,500,000	General
S14	Google Play Store	51	1	303,694	Group
S15	Google Play	1	1	4,442	Individual
S16	Apple AppStore	245	23	7,396,551	General
S18	Apple AppStore Google Play Store	10	2	25,035	Group
S19	Google Play	100	18	5,792	General
S20	Google Play Store	60	Unspecified	21,000	Group
S21	Google Play Store	1	1	1,148,032	Individual
S22	Google Play Store	7	Unspecified	22,815	Individual
S23	Apple AppStore	7	4	59,204	General
S24	Apple AppStore	7	Unspecified	919	General
S25	Google Play Store	8,431	Health apps and non-health apps	383,758	Group

4.3.8. Identified Factor and its Associated Polarity (SQ6)

We identified an initial set of 118 non-unique factors from the 25 publications. First, one researcher analyzed each factor and grouped them according to their name, description, and/or keywords provided in the publication. Some publications classified the same factor into positive and negative ones, such as Aesthetics-Negative and Aesthetics-Positive (Ha and Wagner [S04]), and were grouped into a single factor without polarity. In this first iteration, we ended up with 55 unique factors. One of the authors created a mind map to represent all the factors and presented it to another researcher, an expert in Software Engineering and HCI, to review it. Both discussed the merging process and refined the set of factors. Some factors were complementary to the other, such as “Recommendation” and “Dissuasion.” The former is related to reviews that recommend purchasing or installing the application, while the latter advise against purchase. In this case, we grouped them into the “Recommendation” factor and merged their definition, as both situations (suggestion for acquisition and advise against purchase) are related to a recommendation. We also defined a set of keywords for each factor to characterize them and facilitate overlap identification. We merged factors with overlapping keywords into a unique and broader one. Finally, factors with generic definitions, such as “adjective,” “praise,” and factors that are not informative, such as “work,” defined as reviews that report that the application works without technical description, were removed (factors with grey background). At the end of the process, we identified 31 unique factors (factors with green background). A list with all the original factors and their consolidation can be found in APPENDIX C.

After the consolidation process, we grouped them into categories according to their concept. First, we defined three high-level conceptual categories to group the factors according to the definition of UX presented in Section 4.2.1.4 as follows: App Factors, User Factors, and Context Factors. **App Factors** are related to the app itself, such as its characteristics, functionalities, features, and development. **User factors** are those related to users, such as their profile, needs, and the reasons for their positive or negative evaluations. Finally, **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. Figure 4.10 presents the mapping and merging process of all identified factors with the respective categories.

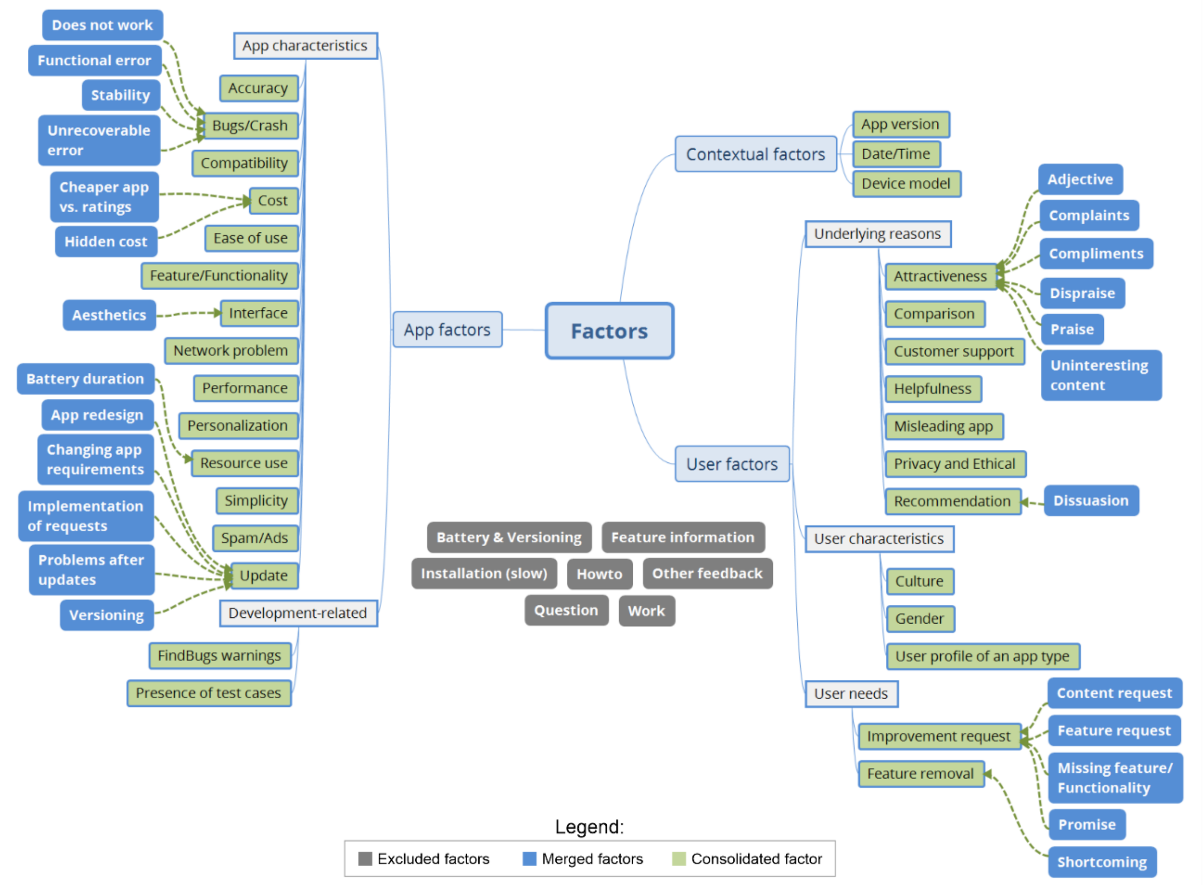


Figure 4.10 – Factors mapping and merging process.

In the next subsections, we present the concept behind each factor and the results of the factors' polarity analysis. We also divided the results into two subsections: i) publications per factor polarity: to identify the polarity of the factors identified by each publication; and ii) factors' polarity: to investigate the polarities that each factor can be associated to, regardless of the publication.

4.3.8.1. Factors' Concept Definition

For each factor, we defined its concept and scope to support researchers and practitioners to better understand it. Some publications did not conceptualize the factor, providing just a set of keywords related to it. For instance, Fu et al. [S03] performed topic modeling by employing LDA to identify users' top complaints by analyzing negative reviews, i.e., with 1 and 2 stars. For each set of keywords from each extracted topic, they derived a name that better represents the concept behind it. For instance, the topic that comprises the words "boring, bad, stupid, waste, don't, hard, make, way, graphic, controls" they named it Attractiveness. The terms imply that the factor is related to users' perceptions and judgments about the app in a given usage situation (indicated by features such as graphics and controls). In this sense, we looked for definitions that would fit this factor. The following definition from Hassenzahl (2018) addresses the idea of the factor: "The user reports experiences with and feelings towards a product in a particular situation into an evaluative judgment." Thus, we set it as the definition of the "Attractiveness" factor. There were also some factors in which the definition was vague. Khalid et al. [S10] provided the following description for the "Privacy and Ethical" factor: "The app invades privacy or is unethical." However, the concept of privacy invasion is lacking. In this sense, we searched for more complete definitions. The work from Ebrahimi et al. (2020) defines privacy invasion in the context of mobile apps as "constant location tracking, unsolicited data collection, or any form of features that are engineered to lure users into sacrificing their privacy in exchange for more personalized services". It defines both privacy issues and app developers' unethical behaviors, reflecting the factor's concept.

To better visualize and organize the data, one of the authors created a mind map with all the available definitions for each selected factor and their respective sources. Through this mind map, we analyzed the definitions to select the adequate one. In some cases, a publication provided a complete definition that conveys the concept of the factor. Thus, we selected it as a default. In other cases, we had to merge the description provided by multiple publications to generate a broad definition to capture all the aspects related to the factor. The 'Update' factor, for instance, had seven other descriptions, as presented in Figure 4.11. We abstracted non-overlapping definitions from each publication and merged them into a unique one. All the process was peer-reviewed by another researcher, an expert in Software Engineering and HCI.

Table 4.8 presents the final definition for each factor extracted in this systematic mapping study. We did not include a definition for “App Version” and “Date/Time” factors, given that they are obtained from the metadata of the reviews, and not through the analysis of the reviews’ content.

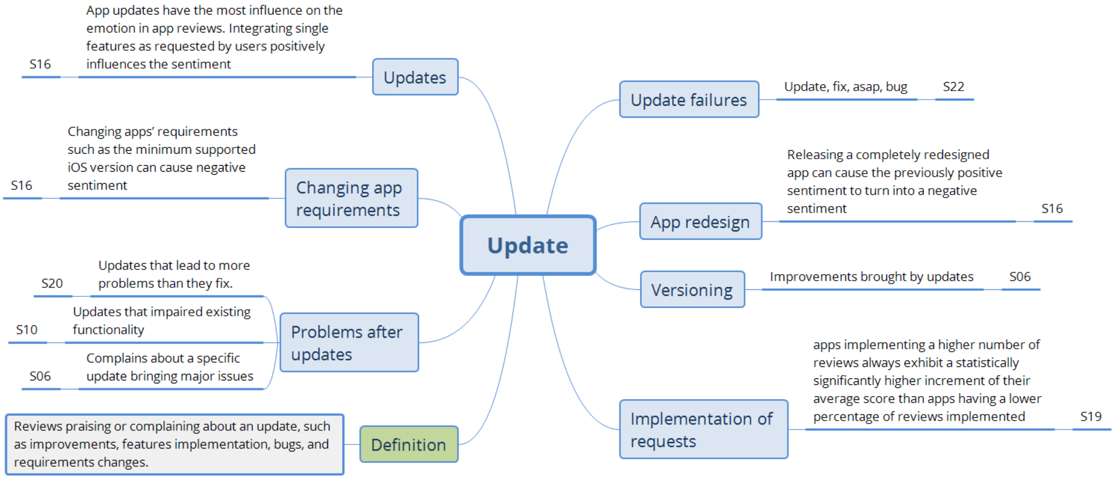

Update
Updates
App updates have the most influence on the emotion in app reviews. Integrating single features as requested by users positively influences the sentiment
S16
Changing app requirements
Changing apps’ requirements such as the minimum supported iOS version can cause negative sentiment
S16
Problems after updates
Updates that lead to more problems than they fix.
S20
Updates that impaired existing functionality
S10
Complains about a specific update bringing major issues
S06
Definition
Reviews praising or complaining about an update, such as improvements, features implementation, bugs, and requirements changes.
Update failures
Update, fix, asap, bug
S22
App redesign
Releasing a completely redesigned app can cause the previously positive sentiment to turn into a negative sentiment
S16
Versioning
Improvements brought by updates
S06
Implementation of requests
apps implementing a higher number of reviews always exhibit a statistically significantly higher increment of their average score than apps having a lower percentage of reviews implemented
S19

Figure 4.11 - Definitions for the 'Update' factor.

Table 4.8 - Definition of the factors extracted in the Systematic Mapping study.

Factor	Definition	Source
Accuracy	The degree to which data has attributes that correctly represent the true value of the intended attribute of a concept or event in a specific context of use.	Fu et al. [S03], ISO/IEC 25012 (2008)
App version	The version of the app the user was using when writing the review.	Goul et al. [S01]
Attractiveness	The user reports experiences with and feelings towards a product in a particular situation into an evaluative judgment.	Fu et al. [S03], Hassenzahl (2018a)
Bugs/Crash	The reviewer writes that the application doesn’t work and provides a technical description, such as it takes too long to load or that it keeps crashing.	Ha and Wagner [S04]
Comparison	Reference to other apps, e.g., for comparison	Pagano and Maalej [S05]
Compatibility	App has problems on a specific device or an OS version	Khalid et al. [S10]
Cost	The reviewer praises the application for being free or, if it is paid, the application/service is worth/not worth the money.	Ha and Wagner [S04], Khalid et al. [S10]
Culture	Differences between the culture of each country considering six dimensions from the Hofstede model: Power Distance;	Guzman et al. [S23]

Factor	Definition	Source
	Individualism vs. Collectivism; Masculinity vs. Femininity; Uncertainty Avoidance; Long-term vs. Short-term Time Orientation; Indulgence vs. Restraint.	
Customer support	Users being satisfied or dissatisfied with the support they received while using apps	[S06]
Date/Time	The date/time when the user wrote the review.	Goul et al. [S01]
Device model	The device the user reported in the review or was obtained through the metadata provided by the app store.	Khalid et al. [S09]
Ease of use	Reviews reporting users' perceptions of the effort related to the usage and the experience of the interaction with the application.	[S07], Weichbroth (2020)
Feature/ Functionality	Reviews that praise or criticize an existing feature (e.g., songs, themes, video quality) or functionality (e.g., upload file, take photo).	[S04]
Feature removal	Complaint about a disliked feature that is degrading the experience	[S10]
FindBugs warnings	The category of warnings from the FindBugs tool: Bad practice, Internationalization, and Performance.	[S13]
Gender	The gender of the user based on the verification of the first name in the generize.io database. Only probabilities higher than 95% for "male" or "female" are considered. Lower probabilities are assigned the "unisex" label. Names not occurring in the database are given the "unclear" label.	[S24]
Helpfulness	Comprises topics related to descriptions of the app in action. These are helpfulness, which captures use cases where the application has proven helpful, and feature information, including descriptions of application features and user interface.	[S05], [S16]
Improvement request	Requests improvement (e.g., app is slow) or the addition of new features or content	[S05]
Interface	Reviews that describe the application's overall look or interface, including images, color scheme, icons, and text	[S04]
Misleading app description	User reports that the description of the app and all the visuals associated with it does not	Iacob et al. [S21], Panosian (2017)

Factor	Definition	Source
	accurately convey the functions and features of the app	
Network problem	The app had trouble with the network or responded slowly.	[S10]
Performance	The app is slow to respond to input, or laggy overall.	[S10]
Personalization	The extent to which the Web site can be customized to the needs of individual customers. Customization also provides flexibility and control regarding the content and organization of the information they want (Huang, 2002) and facilitates interactivity (Schubert & Selz, 1997)	Tarafdar and Zhang (2005)
Presence of test cases	Apps with automated tests in which the adopted test-to-code-ratio was 1 line of test code to 10 lines of production code (i.e., 1:10): a ratio of 1:10 indicates that, for every line of test code, there are 10 lines of production code.	[S20]
Privacy and Ethical	Reviews reporting app developers' unethical actions (e.g., unethical business practices or selling users' personal data) or that the app requests information that may invade users' privacy, such as personal information, constant location-tracking, unsolicited usage data collection or any form of features that are engineered to lure users into sacrificing their privacy in exchange for more personalized services.	[S10], Ebrahimi et al. (2020)
Recommendation	The user suggests acquisition or advises against purchasing or downloading the app	[S05]
Resource Use	The app consumes or does not consume too much battery/memory.	[S10]
Simplicity	The degree of being easy to understand or being uncomplicated in form or design, described by such characteristics as the number of menu levels, the number of performed gestures to reach a destination object, and the duration of searching a button to perform a specific function.	[S14], Weichbroth (2020)
Spam/Ads	Review complains about the number and content of ads in the application or says that	[S03], [S04]

Factor	Definition	Source
	there weren’t too many ads in the application or wouldn’t mind having a free application that contained ads.	
Update	Reviews that praise or complain about an update, such as improvements, features implementation, bugs, and requirements changes.	[S06], [S10], [S16], [S20]
User profile of an app type	Users’ profiles can affect ratings. Healthcare app users, for instance, tend to be less critical towards errors appearing in such apps, leading to more positive ratings.	Nicolai et al. [S25]

4.3.8.2. Publication per Factor Polarity

We identified 17 publications that reported factors associated to negative reviews (Figure 4.12). Some works, such as Fu et al. [S03], focused on investigating only negative reviews. They investigated more than 13 million user reviews of 171 thousand Android apps and extracted 10 factors that were associated with negative reviews: attractiveness, stability, accuracy, compatibility, connectivity, cost, telephony, picture, media, and spam. They grouped the applications into two main categories and their respective subcategories: game (arcade, puzzle, sports, etc.) and general application (communication, education, social, etc.). For each subcategory, they identified the top three factors associated with negative reviews. For games, all the subcategories had attractiveness, stability, and cost as the main factors associated with negative reviews. By contrast, the main factors for each subcategory of general applications varied, not being possible to identify a pattern.

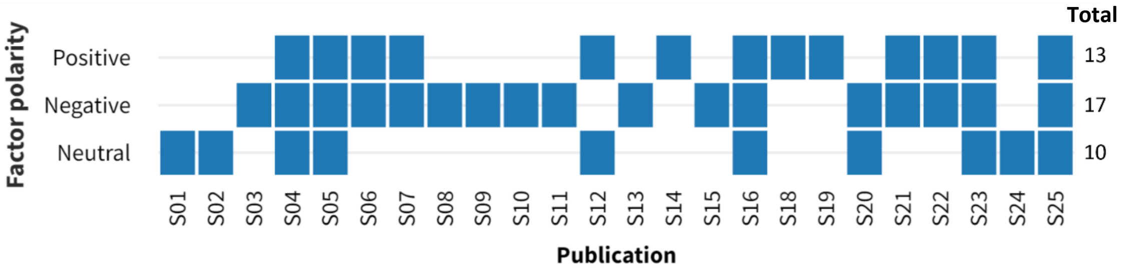


Figure 4.12 - Number of publications by the associated polarity of the factors they addressed.

There were 13 publications that reported factors associated with positive reviews. A study conducted by Nicolai et al. [S25], for instance, revealed that users of healthcare applications tend to be more positive when describing and reporting failures than users of other types of apps. The "problem discovery" and "problem reporting" categories, for instance, had 54% and 62% of positive reviews, respectively, from healthcare apps users. By contrast, these categories did not have any positive reviews from non-healthcare apps users. According to the authors, this is likely due to their willingness to (i) be proactive concerning apps that help their life and social activities and (ii) drive developers toward the resolution of problems rather than blame them for missing functionalities.

Finally, 10 publications presented factors that did not affect either positively nor negatively. For instance, Guzman and Paredes-Rojas [S24] indicated that gender does not have any influence on the variables they analyzed: user's rating, time to post a comment after an app release, and review's length, sentiment, and content. In another study, Goul et al. [S20] did not find any significant difference on ratings of apps with and without test cases.

4.3.8.3. Factors' Polarity

We analyzed each of the 31 factors by mapping their associated polarity. FiguresFigure 4.13,Figure 4.14, and Figure 4.15 presents the factors according to the polarities they are associated with. Each factor can be associated with one or more publications that analyzed it according to different scopes and produced results applicable to the individual, group, or general contexts.

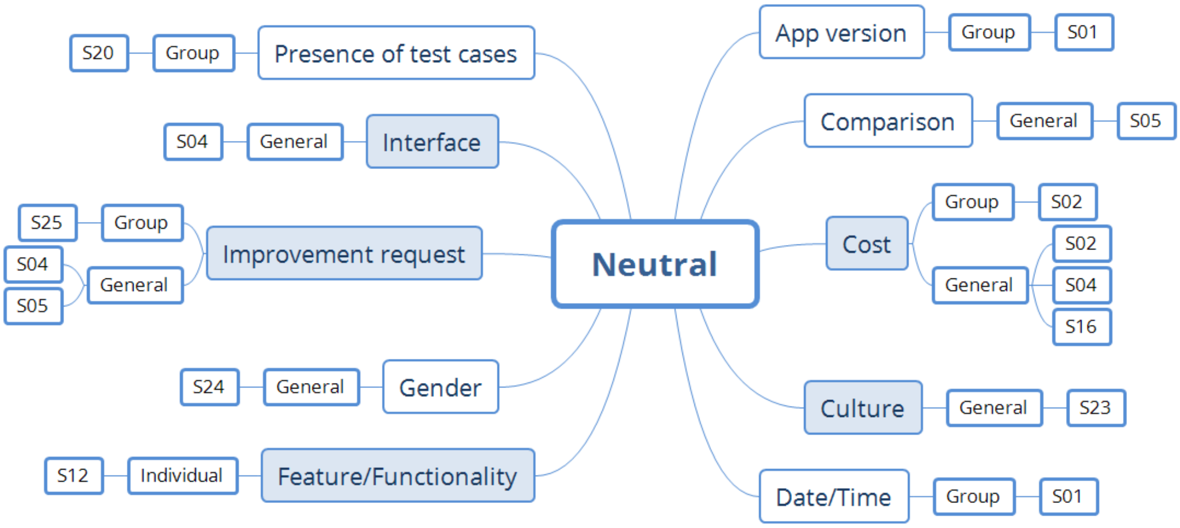


Figure 4.13 - Factors associated to neutral evaluations.

Most of these factors were exclusively associated with negative reviews, such as Compatibility (Fu et al. [S03], Khalid et al. [S10], Luiz et al. [S22]), Privacy and ethical (Khalid et al. [S10]), and Spam/ads (Fu et al. [S03], Luiz et al. [S22]). Others were only associated with positive reviews, such as User profile of an app type (Nicolai et al. [S25]), Helpfulness (Pagano and Maalej [S05], Martens and Johann [S16]), Ease of Use (Guzman and Maalej [S07]), and Simplicity (Mohan et al. [S14]). A smaller portion was associated with neither positive nor negative reviews, such as the Presence of test cases (Durelli et al. [S20]), Gender (Guzman and Paredes-Rojas [S24]), and App version (Goul et al. [S01]).

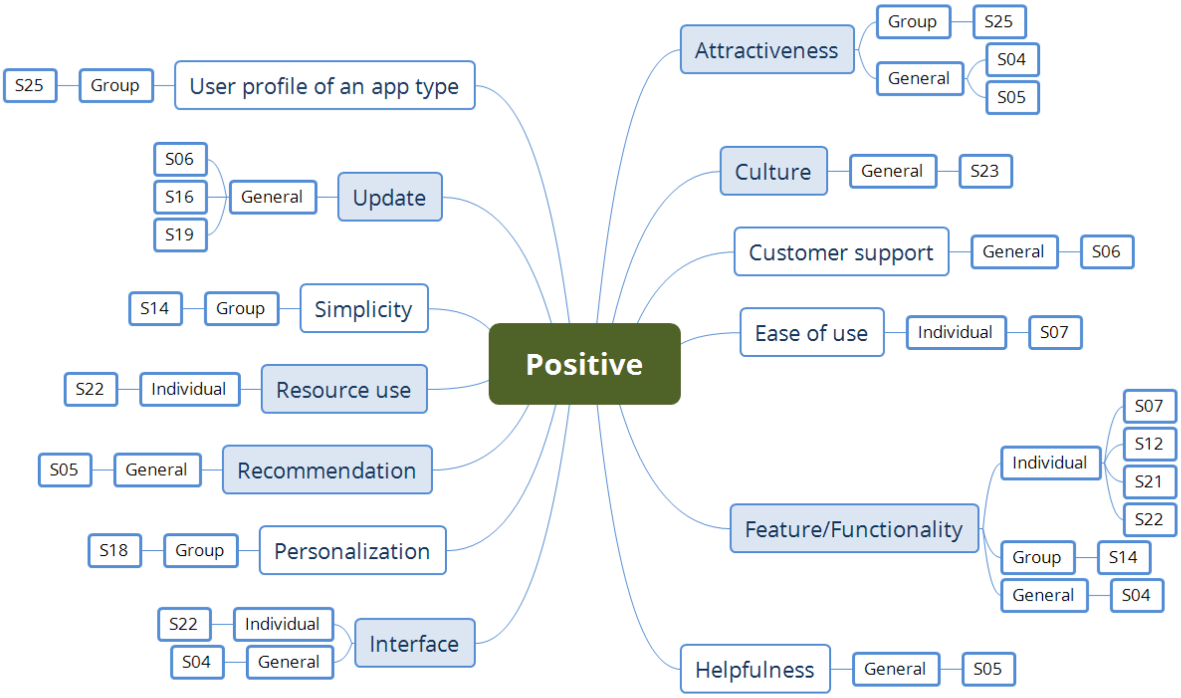


Figure 4.14 - Factors associated to positive evaluations.

There were also factors associated to more than one polarity (factors with blue background): *Attractiveness*, *Bugs/Crash*, *Cost*, *Feature/Functionality*, *Improvement Request*, *Interface*, *Recommendation*, *Resource Use*, and *Update*. We analyzed these factors to understand the reasons for the contradictory results better.

The main reason is the **difference in data analysis**. Regarding *Cost*, all publications that investigated the correlation between price and user reviews found that the correlation was not significant, i.e., a cheaper or more expensive app will not necessarily lead to a more positive or negative evaluation (Harman et al. [S02], Iacob et al. [S06], Martens and Johann [S16])). In turn, some publications focused only on negative reviews (Fu et al. [S03], Khalid et al. [S10]) or that divided them into positive and negative reviews (Ha and Wagner [S04]), leading to contradictory results due to the differences in the analysis. *Interface* was associated with all three polarities and addressed by three publications with different analyses. One of them analyzed the reviews at the app level (Luiz et al. [S22]), one focused only on negative reviews

(Khalid et al. [S10]), and one divided it into positive and negative interface evaluations (Ha and Wagner [S04]).

Factor's neutral nature: all these factors were already associated with either negative or positive polarity according to the publications they were extracted. Some publications divided the same factor into positive and negative polarities to classify and filter the reviews. Pagano and Maalej [S05], for instance, presented two generic factors (*Praise* and *Dispraise*), both related to users' appreciation towards the app (*Attractiveness*) but with opposing polarities. They also separated reviews recommending the acquisition of the app from those dissuading users not to acquire it, both related to *Recommendation* factor. Ha and Wagner [S04] also classified the reviews related to interface overall look into Aesthetics-positive and Aesthetics-negative. In addition to these situations, the variation was more frequent when analyzing the factor at the app level. *Feature/Functionality*, for instance, was the factor associated with all three polarities. This variation is because five out of ten publications analyzed the reviews at the app level, extracting specific features from each app. Each feature/functionality can be evaluated either positively or negatively by users according to the app, thus leading to many variations in the results.

Regarding *Resource Use*, two publications identified that battery drain is associated with negative evaluations (Khalid et al. [S10], Durelli et al. [S20]), while one publication associated the small impact on battery duration to positive evaluations in reviews of some specific apps (Luiz et al. [S22]). Regarding *Culture* [S23], the ratings and sentiments varied in each country analyzed. These variations occurred due to different cultural values from two out of six dimensions of the Hofstede's model (Geert and Jan, 1991): Power Distance (degree to which members of the country accept and expect that power is distributed equally) and Indulgence (the extent to which a society expresses their wants and impulses). Indulgence correlated positively with user ratings, while Power Distance correlated negatively. According to the authors, higher indulgent countries tended to provide more positive ratings, while countries with lower Power Distance tended to provide more negative ratings.

Regarding *Improvement Request*, four out of six publications (Pagano and Maalej [S05], Khalid et al. [S10], Li et al. [S21], Luiz et al. [S22]) associated it with negative reviews, indicating that users usually penalize the app due to the lack of features or functionalities. The other two publications (Ha and Wagner [S04] and Nicolai et al. [S25]) associated it to neutral evaluations. Regarding the work of Ha and Wagner [S04], the neutral evaluation may be due to the small sample size, as they obtained 556 reviews from 59 apps, which results in less than 10 reviews analyzed per app. Regarding the work of Nicolai et al. [S25], the neutral evaluation can be explained by the target population of the apps analyzed (healthcare users), as they tended to be more positive in general. Regarding *Updates*, most of the publications related it to negative evaluations due to problems brought by app updates, such as changes in app requirements (e.g. required a different OS version to be installed) and app redesigns that changes users' workflow (Iacob et al. [S06], Khalid et al. [S10], Martens and Johann [S16], Durelli et al. [S20]).

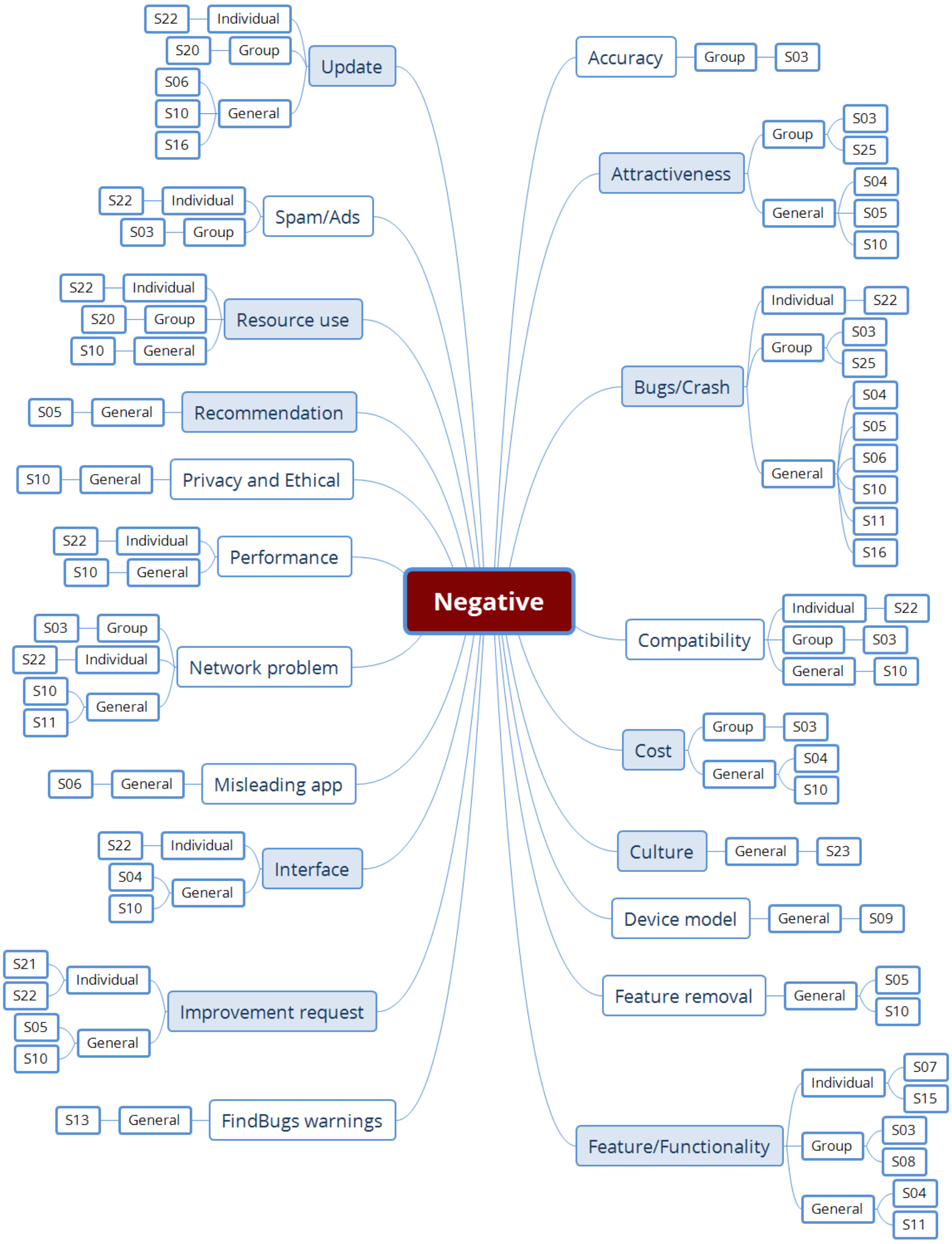


Figure 4.15 - Factors associated to negative evaluations.

Dynamic nature of users' experiences, expectations, and needs: another possibility to explain the contradictory results is related to changes in users' experiences, expectations, and needs over time. According to Law and Van Schaik (2010), user expectation and affect evolve dynamically in the long run. Aspects that were perceived as a product's differential in the past may turn into an aspect that is considered mandatory nowadays. In the work of Ha and Wagner [S04], for instance, the authors found that a good-looking and usable interface will not imply better ratings, but a bad interface or not usable one results in a decrease of users' ratings. It indicates that users are more demanding, requiring the software product to be good-looking and usable, increasing its weight in users' evaluations. Thus, it is essential to develop approaches that capture the variations of the weights of these factors over time.

4.3.9. Factor Influence Analysis (SQ7)

Sixteen out of 25 publications analyzed the influence of the factors on users' sentiments or ratings. Some studies employed statistical tests to investigate whether the differences between different groups are significant, such as apps with and without test cases (Durelli et al. [S20]), device model (Khalid et al. [S09]), and the influence of gender (Guzman and Paredes-Rojas [S24]) and culture (Guzman et al. [S23]). Other studies performed some frequency/distribution analysis, for instance, to investigate the impact of each factor by analyzing their frequency according to the number of star ratings (Pagano and Maalej [S05]) and the ratio between one and two-star ratings (Khalid et al. [S10]). Finally, some publications performed correlation analysis to investigate the relationship between different variables, such as date/time and app version with ratings (Goul et al. [S01]), price and ratings (Harman et al. [S02]), and between emotions, ratings, and price (Martens and Johann [S16]). To better understand the reasons behind positive, negative, and neutral evaluations, we compiled the findings from each factor across publications (see Table 4.9).

4.3.10. Results from the Secondary Study

One of the accepted publications (S17) returned by the search engines was a systematic mapping study conducted by Genc-Nayebi and Abran [S17]. In this work, the authors aimed to address publications that proposed solutions for mining app store user reviews, reported challenges and unsolved problems in the domain, and contributions for software requirements and evolution. To do so, the authors defined five research questions to get information about: i) the data mining techniques employed; ii) remedies for the domain dependency challenge; iii) review usefulness criteria; iv) spam identification; and v) extracted features. According to the authors, the search string returned more than 500 publications, but they did not provide a precise

Table 4.9 - Findings regarding each factor identified.

	Polarity			Analysis Scope				
Consolidated Factor	**Pos**	**Neg**	**Neu**	**Ind**	**Gro**	**Gen**	**Papers**	**Findings**
Accuracy		X			X		S03	**Group:** complaints related to lack of accuracy on finding information and location. It is common in the following categories: Book & Reference, Lifestyle, Productivity, Transportation, Travel, and Weather.
App version			X		X		S01	**Group:** There was no correlation between app version and ratings for business intelligence apps.
Attractiveness	X	X			X	X	S03, S04, S05, S10, S25	**General:** Reviews including emotional expressions result in more positive or negative evaluations than reviews without emotional expressions (S04). Unappealing content usually leads to negative evaluations (S10). **Group:** healthcare user apps are usually less satisfied in general compared to users of non-health apps (S25). Content is key success for mobile games (S03).
Bugs/Crash		X				X	S03, S04, S05, S06, S10, S11, S16, S22, S25	**General:** bugs/crash was the third most negative factor (S05), with most complains related to functional errors, especially location and authentication issues (S10). Users reporting bug-related issues tend to write the reasons why the app does not work and evaluate it very negatively (S04). Around 18% of post-update reviews complained about frequent crashing (S10). Although severe bugs greatly impact users' experience, minor bugs seem not to impact so much on user ratings (S06). **Group:** users of healthcare apps expect developers to improve the app and help them take care of their health. As consequence, they are less critical towards app errors compared to users of non-healthcare apps (S25). All categories of games suffered from stability problems that resulted in negative evaluations (S03).
Comparison			X			X	S05	The authors did not further detail the factor, for example, what aspects are considered in comparisons (S05).
Compatibility		X		X	X	X	S03, S10, S22	Compatibility issues led to negative evaluations in general, category, and individual analysis (S03. S10, S22). **Individual:** the use of PicsArt app on tablets causes na undesired increase in users' dissatisfaction.

	Polarity			**Analysis Scope**				
Consolidated Factor	**Pos**	**Neg**	**Neu**	**Ind**	**Gro**	**Gen**	**Papers**	**Findings**
Cost		X	X		X	X	S02, S03, S04, S06, S10, S16	**General:** diverging results identified. Publication S04 suggests that negative reviews about the cost can decrease ratings, but positive reviews do not impact the overall rating. However, its sample size is small. Publication S02, in turn, suggests that price does not have influence in the number of downloads or ratings. The greater number of apps (32,108) and the rigor of the analysis using statistical tests strengthen the findings of this publication compared to S04. This finding is supported by publication S16, in which the authors did not find any correlation between price and emotions, claiming that users willing to post a review provide feedback due to intrinsic motivation. **Group:** cost is one of the three main reasons why users dislike a mobile game (S03). However, users of such apps seem to be more tolerant regarding prices, as paid apps have more complaints about their prices than paid games (S03). Cheap apps are rarely reported as worth the price (S06), and users also do not like apps that claim to be free but ask them to pay to get access to some features (S10).
Culture	X	X	X			X	S23	**General:** the rating of the reviews of a specific country positively correlates with the Indulgence of the country and negatively correlates with its Power Distance.
Customer support	X					X	S06	**General:** the majority of the users (61.58%) are positive regarding customer/developer support, resulting in positive evaluations.
Date/Time			X		X		S01	**Group:** Date/Time were not correlated with sentiment of reviews of productivity apps, but it is not further analyzed.
Device		X				X	S09	**General:** the results revealed the importance of analyzing reviews according to the devices used by users, as some of them tend to receive worse ratings than others due to specific problems.
Ease of Use	X			X			S07	**Individual:** for the Pinterest app, its ease of use was the most positively evaluated aspect (S07).
Feature removal		X				X	S05, S10	**General:** features users do not like have high influence on user ratings (S05). It was the third most complained factor, mostly leading to reviews with 1 star rating (S10).

	Polarity			Analysis Scope				
Consolidated Factor	**Pos**	**Neg**	**Neu**	**Ind**	**Gro**	**Gen**	**Papers**	**Findings**
Feature/Functionality	X	X	X	X	X	X	S03, S04, S07, S08, S11, S12, S14, S15, S21, S22	Specific features or functionality can lead to either positive, negative, or neutral ratings according to the app or category. For instance, regarding mobile banking apps, features like money transfer, card payments, account summary, and ease of use were associated with positive evaluations (S14). Problems with media (watching, listening, and recording) negatively affected the Entertainment, Media & Video, and Music & Audio categories of apps (S03). Problems with Pictures (see, save, and upload photos) negatively affected Comics, Media & Video, Personalization, and Photography categories (S03).
FindBugs warnings		X				X	S13	**General:** code warnings related to Bad practice (i.e., violation of essential coding practices, for example, equals problems, dropped exceptions, and misuse of finalize), Internationalization (misuse of encoding characters) and performance (slow code) are correlated with low ratings.
Gender			X			X	S24	**General:** there was no significant difference in ratings between males and females. Regarding sentiment, there was difference in gender, but not significant.
Helpfulness	X					X	S05	**General:** the second most popular factor and the second most positive, commonly associated with reviews praising the app. Its polarity increases when associated with recommendation.
Improvement request		X	X	X	X	X	S04, S05, S10, S21, S22, S25	**General:** In general, feature requests are associated with negative reviews (S05, S10, S21, S22), indicating that missing features affect users' evaluations. However, on average, reviews associated with this factor still lies above the middle (3 stars), indicating that it does not have a great impact on ratings (S05). Content request was the least critical requirements feedback, with small impact on user ratings (S05). Missing features have less impact when users already liked the app (S04) and praise it in the reviews (S05), resulting in more positive evaluations. **Group:** healthcare apps users request more features than users of non-healthcare apps. In turn, they tend to not evaluate the app negatively due to missing features (S05).

	Polarity			Analysis Scope				
Consolidated Factor	**Pos**	**Neg**	**Neu**	**Ind**	**Gro**	**Gen**	**Papers**	**Findings**
Interface	X	X	X	X		X	S04, S10, S22	**Individual:** regarding WhatsApp, negative reviews were from users requesting features related to contact options and status (S21), and upgrade of themes (S22). **General:** in general, reviews complaining about the interface of the app can decrease ratings (S04, S10). Conversely, reviews praising the interface does not lead to more positive evaluations (S04). **Individual:** reviews praising the interface of the app resulted in positive evaluations for DropBox and PicsArt apps (S22).
Misleading app description		X				X	S21	**General:** misleading app descriptions are common (21.85%) and can result in negative evaluations (S21).
Network problems		X		X	X	X	S03, S10, S11, S22	In general, network issues, such as wi-fi and mobile network problems, connection errors, login issues, and slow connections, leads to negative evaluations.
Performance		X		X		X	S10, S22	In general, problems related to slow responses to user input or overall performance can lead to negative evaluations.
Personalization	X				X		S18	**Group:** reviews reporting personalization possibilities were positive in educational apps.
Presence of test cases			X		X		S20	**Group:** The median of apps with and without automated tests did not differ significantly, indicating that it does not affect users' perceptions of the experience with the app.
Privacy and Ethical		X				X	S10	**General:** it was the most critical factor. Users are bothered by privacy invasion and the app developer’s unethical actions (for example, unethical business practices or selling the user’s personal data).
Recommendation	X	X				X	S05	**General:** Users can either recommend to or dissuade other users. Reviews recommending the app are positive. In turn, reviews dissuading other users are very negative. The combination of bugs and dissuasion results in the lowest average rating (S05).
Resource use	X	X		X	X	X	S10, S22	In general, resource heavy apps (e.g., consumes much battery, memory/storage) are negatively evaluated (S10, S20, S22). In turn, users also recognize when an app is energy-efficient, leading to very positive evaluations (S22).

	Polarity			**Analysis Scope**				
Consolidated Factor	**Pos**	**Neg**	**Neu**	**Ind**	**Gro**	**Gen**	**Papers**	**Findings**
Simplicity	X				X		S14	**Group:** for mobile banking apps, simplicity, friendliness, and easiness were the main reasons for the positive sentiment (S14).
Spam/Ads		X		X	X		S03, S22	In general, ads are considered annoying, having negative influence on ratings (S03, S22). It was the third most complained factor in the "Personalization" apps category (S03). In the Angrybirds game, this factor resulted in the lowest ratings (S22).
Update	X	X		X	X	X	S06, S10, S16, S19, S20, S22	**General:** updates implementing requests from users result in positive reviews (S06, S16, S19). Conversely, releasing a completely redesigned app can cause the previously positive sentiment to turn into a negative sentiment (S16). Many of the complaints are related to functional errors (S10) and usability issues (S06) after recent updates. **Group:** problems related to updates are the main source of negative reviews in apps with test suites (S20). **Individual:** the worst ratings from Evernote app was due to app update failures (S22).
User profile of an app type	X				X		S25	**Group:** Healthcare apps' customers tend to be more positive when describing and reporting failures than users of other apps. Similarly, they try to recommend possible solutions to those errors in a more polite way.

number. Among them, 63 publications remained as start set for the backward and forward snowballing process, which resulted in 45 research papers. However, the authors did not provide details of how many publications they excluded. At the end of the process, the authors selected 24 primary studies after applying the inclusion and exclusion criteria.

The authors listed five main findings: i) most of the studies were exploratory, based on manual classification and correlation analysis; ii) the approaches to extract app features do not consider the nature of app store reviews, such as short length, unstructured phrases, colloquial language, and abundant information; iii) users and developers request a different type of information, i.e., the former is interested in the experience of other users, while the latter seeks to improve the app quality by addressing missing requirements, features, and user experience information; iv) external sources of information, such as tweets, blogs, and code repositories could be used to enrich the data; v) identifying useful reviews, as well as spam and fake reviews are one of the biggest problems in the domain.

The main difference between their work with our systematic mapping study is that they focused more on addressing general questions related to app store reviews mining, such as the domain dependency challenge (i.e., the issue related to a classifier that is trained in a given domain and performs poorly when applied into another), reviews aggregation, and spam detection. Although they mention factors in some parts of the paper, they did not address their relationship with users' ratings and sentiments. In contrast, our research was conducted from the UX perspective by investigating what factors can affect the UX conveyed by mobile applications, what analyses were carried out, and the scope of the analysis. We also analyzed the factors we extracted and aggregated them into a comprehensive set. With this study, we take one step towards understanding the factors that can affect UX to advancing the research on app store reviews mining and HCI fields.

4.3.11. Results Summary

Table 4.10 presents an overview of the results based on the primary studies' counting into each sub-question. Each paper can be assigned to more than one answer in each sub-question, except in SQ4. Thus, the number of papers assigned to each sub-question can be greater than the number of primary studies included in this systematic mapping study (24). The complete list of the publications and the respective mapping to each SQ are available in APPENDIX B.

One of the accepted publications (S06) returned by the search engines was a systematic mapping study conducted by Genc-Nayebi and Abran (2017) to address app stores opinion mining studies.

Table 4.10 – Overview of the results for each research sub-question.

Sub-question	Answer	Number of publications
SQ1. Dataset source	Google Play Store	15
	Apple AppStore	9
	Other	2
SQ2. Extracted information	Reviews	23
	Ratings	18
	App information	7
	Other	3
SQ3. Analysis methods	Descriptive statistics	12
	Sentiment analysis	11
	Statistical tests	9
	Manual analysis	6
	Topic modeling	5
	Other	4
SQ4. Data categorization	Yes	12
	No	12
SQ5. Scope of the analysis	General	12
	Group	8
	Individual	5
SQ6. Factor's associated polarity	Negative	17
	Positive	13
	Neutral	11
SQ7. Factor's influence analysis	Yes	16
	No	8

4.4. DISCUSSION

In this subsection, we present the main findings from this systematic mapping study. We also highlight and discuss the implications for researchers and practitioners.

What are the influencing factors on users' evaluations in app stores reviews, and how do they affect the evaluation? Regarding our research question, we identified 31 unique factors that can affect users' evaluations. While factors associated exclusively with negative reviews are more related to features and functionality issues (e.g., Performance, Compatibility, Accuracy, Feature removal, Network problem), positive factors are more related to general perceptions and human aspects (e.g., Helpfulness, Ease of Use, Customer Support, User profile of app type, Culture). It indicates that dissatisfied users tend to provide details about the functionalities and aspects they are not happy with. By contrast, when giving positive reviews, they tend to describe the app's overall qualities and aspects.

Some factors can have different effects on users' evaluations according to their polarity. For instance, while negative reviews on the app's cost and the interface can decrease ratings, positive ones do not impact the overall rating (S04). Some factors appeared more frequently in a given type of app. Attractiveness, Stability, and Cost were the three top factors in mobile games (S03). Other findings are related to the impact of different factors. Privacy and Ethical were the most critical factor, with the greatest ratio of 1-to-2-star ratings (S10), as well as the Spam/Ads factor, which led to the lowest ratings for a mobile game (S22). There was also a factor in which its effects depended on different variables. Regarding Update factor, while implementing small improvements requested by users can increase ratings (S16, S19), a completely redesigned interface can lead to dissatisfaction (S16). Usability issues, update problems, and broken functionalities due to a new release are also other complaints that can result in negative evaluations. In this sense, developers should be careful when updating their apps. Developers must monitor the number of positive and negative reviews related to this factor over time, especially after releasing an update.

From practitioners' perspectives, all this information provides clues of factors that they should consider when developing or improving their apps. The identification of the impact of these factors could help practitioners to decide which of them to prioritize for improving the UX the app conveys. From the academic perspective, researchers could attribute different weights to the factors when evaluating the UX of mobile applications or designing weighted UX evaluation methods, such as the weighted heuristics proposed by Lynch et al. (2013) to evaluate websites for older adults. Researchers can also support the software development process by proposing approaches that automatically analyze user reviews to identify these influencing factors and their effect on the app developed by the company .

Lack of comparative studies between different app stores (SQ1): the reviews were obtained only from Google Play Store and Apple AppStore. This result was expected, as Android and iOS are currently the two most widely used mobile operating systems. However, there is a lack of studies that analyzed reviews from both app stores. This makes it difficult for researchers and practitioners to identify particularities of applications targeted to different operating systems and evaluate whether the findings from one app store (or from an app designed for a particular OS) apply to the other. By comparing different OS, software engineers can identify whether a given bug or interaction problem may be due to the platform or the app itself. It will also make it possible, for instance, to investigate whether needs, expectations, and factors that affect UX differ between groups, allowing developers to design products that are personalized to different target populations.

Ratings are the most common variable when analyzing user reviews (SQ2): most of the studies extracted ratings from user reviews to investigate the influence of different variables, such as gender (S23) and features (S15, S19). Ratings were also employed to identify the impact of factors according to the proportion of negative reviews, i.e., with 1 and 2 stars (S05, S10). By contrast, publications S03 and S22 performed sentiment analysis to identify inconsistent reviews. The results indicated that star ratings are, in some cases, inconsistent with the sentiment identified in the reviews. Such results highlight the possibility of developing approaches to filter such inconsistencies and reduce noise to get more precise results when analyzing the impact of factors on UX.

Various methods were employed to analyze user reviews (SQ3): different analysis methods were employed according to the study's goals. Descriptive statistics were primarily used to identify relationships between variables and identify their impact on ratings and sentiment. The considerable number of studies applying sentiment analysis techniques indicate the need to get further information from reviews to understand UX better, considering that the star ratings only reflect users' overall perception of the experience with the app and not of specific aspects. Moreover, inconsistencies between ratings and sentiment reinforce the need for using complementary approaches to get more reliable results. However, we did not identify a method specifically designed to analyze app store reviews. In contrast to reviews from other online stores, mobile app store reviews are generally short in length, given that they are written and submitted from mobile devices, on which typing is not easy (Fu et al., 2013). It makes the analysis harder, as less data is available for processing per review. Moreover, according to Martens and Johann (2017), some words considered negative in the software engineering domain may not necessarily be negative by their nature. Consider, for example, the following fictitious user review: "The app has many bugs and crashes all the time on my phone." This review would probably receive a neutral sentiment when analyzed by sentiment analysis methods designed for general purposes. By analyzing this review with SentiStrength[5], for instance, the output was a neutral sentiment. However, words like bug and crash have a very negative connotation in the software engineering domain (Martens and Johann, 2017), reinforcing the importance of employing and developing methods that are adjusted to the particularities of the domain. Statistical tests were applied specifically in studies comparing

[5] http://sentistrength.wlv.ac.uk/

groups of apps. Considering that the rating distribution is skewed among apps (Hu et al., 2018), such publications employed non-parametric tests, such as Mann-Whitney, Chi-Square, and Wilcoxon Rank-sum. Regarding Manual Analysis, most of the publications reported using coding processes to analyze a sample of reviews and group them into categories. However, they did not develop automatic approaches to classify the other reviews, missing the opportunity for deeper analyses. Publications using automatic approaches adopted topic modeling to identify groups of related terms. A drawback of the approaches employed in these studies is that the outcomes are lists of terms, which increases the cognitive load to interpret them.

Little focus on UX and much focus on features/functionalities (SQ4): we identified many publications that categorized the data to analyze it better, most of them considering one or more aspects related to UX, such as performance and aesthetics. However, we did not find any publication that analyzed the reviews in the light of UX theory. Most of the factors were related to functionalities and use of the app, such as performance, battery, and bugs. Some publications mention the term "user experience" along with the paper, and two publications even have a category named as such (Pagano and Maalej [S05], Martens and Johann [S16]). However, they associate this category to reviews that describe a concrete feature or user interface in a scenario that the application has proven helpful. Despite being named "user experience," it does not have a relation to any of the definitions and concepts of UX, but with "descriptions of the app in action" (Pagano and Maalej, 2013). Although many publications considered users' emotions in their analysis, this information was only used to identify their opinions' polarity and extract the main topics associated with each polarity, which may explain the lack of subjective factors related to users' emotions and feelings. Moreover, there was no consolidated taxonomy to categorize the reviews, leading to various categories, many of them without a clear definition of their concept.

Low representativeness weakens the results (SQ5): initial studies involving manual analysis were conducted in small datasets. For instance, Ha and Wagner (S04) analyzed only 556 reviews of 59 different apps from 30 categories. In turn, Iacob et al. (S06) analyzed 3,279 reviews of 161 apps from 6 categories. Both works focused on classifying the reviews into categories derived through manual analysis. Although Ha and Wagner (S04) performed statistical tests to strengthen their findings and found statistically significant results, the small sample reduces the study's statistical power. The low app/reviews ratio also weakens the sample's representativeness to draw conclusions on the factors considered in such studies.

Researchers could employ machine learning techniques to automatically classify and analyze larger datasets to confirm such findings.

Few studies analyzing the particularities of different groups of apps (SQ5): most of the studies analyzed the data in a general context by gathering thousands of reviews of different categories of apps. However, such analyzes only provide an overview of the reviews from the app store, hindering the identification of the particularities of each type of app. Although we identified many studies that analyzed the data in groups, most of them did not compare these different groups of apps. For instance, Nicolai et al. (S25) identified that users of healthcare applications tend to be more tolerant to bugs and lack of functionalities compared to users of other types of applications. By making comparisons, it would be possible, for instance, to identify what factors users are more concerned about according to the group the app belongs to, which may help defining which factors to prioritize during the development process.

Greater number of negative factors (SQ6): the results revealed more negative factors than positives and neutrals. Many publications focused on investigating negative reviews for app improvement purposes, given that positive reviews usually do not point out problems and improvement requests. The negative factors highlight which aspects developers should consider when developing new versions or new apps to improve UX and avoid negative evaluations. By analyzing these factors, we can suggest developers prioritize the correction of bugs (Bugs/Crash), also considering compatibility issues (Compatibility) and the devices associated with them (Device). The description of the app should be clear and reflect the actual functionalities (Misleading app), avoid using advertisements (Spam/Ads), and request non-essential user information that could result in privacy invasion (Privacy and Ethical). Developers should also pay attention to the app’s performance (Performance) and connectivity issues (Network problems) to provide a better experience.

Impact of the factors on users’ sentiments and ratings unclear (SQ7): half of the studies only presented the factors and their associated polarity, while the other half investigated the influence of each factor on users’ sentiments and ratings by performing correlation analysis, statistical tests, or frequency/distribution analysis. Although they indicate whether the factor influence or not on users’ ratings and sentiments, the weight of each factor remains unclear. The publications that performed statistical tests did not present the effect size, an important measure to assess the magnitude of the effect of each factor. In contrast, other publications performed only simple descriptive statistics by analyzing the frequency and distribution of the factors. By identifying the magnitude of the effect of these factors, developers could focus their efforts on aspects that most negatively impact users’ perceptions about their product while maintaining or improving the aspects that are evaluated positively in future releases.

4.4.1. Challenges and Open Issues

We identified the following challenges and open issues:

Impact of the factors unclear: it remains an open issue and is also one of the main challenges. In addition to the absence of the effect size in studies performing statistical tests, there are other challenges to estimate the impact of each factor. For instance, while many factors can be present in a single review, a review has only one rating, making it hard to identify which affected more positively or negatively the user’s evaluation. This issue could be mitigated by analyzing the sentiment of the reviews at the aspect level (aspect-based sentiment analysis). Most of the publications identified in our systematic mapping that performed multilabel classification (e.g., S04, S05, S10) analyzed the factors by grouping reviews assigned to the same factor and calculating the average rating. Such an approach can bias the results, as multi-labeled reviews are considered in analyzing different factors. Future research directions could include developing approaches to analyze the reviews at the aspect level to identify the sentiment associated with each factor within a review and identify their impact on users’ ratings, similar to the one proposed by Wang et al. (2010) for analyzing hotel reviews.

Longitudinal perspective of the factors: only three publications analyzed the reviews from a longitudinal perspective. Xiaozhou et al. (S21) analyzed the variation of users’ sentiment after different WhatsApp releases and identified polarity changes after a given update. In turn, Martens and Johann (S16) analyzed the variation in the sentiment of users of Bank of America and Gmail apps. They identified some emotion drops in both apps and found issues related to changes in requirements and features. Finally, Vu et al. (S11) performed a trend analysis to identify potential problems by comparing the number of occurrences of a given keyword over time and the moving average. Analyzing the factors from a longitudinal perspective is crucial to identify trends and understand UX better, given that it is dynamic and changes over time. Factors that were important in the past may not be so relevant in the present. Considering the dynamic nature of UX, it is crucial to investigate the effect of the factors over time.

4.5. SUMMARY

This systematic mapping study aimed to answer the following research question: “ *What are the UX-related factors that influence users’ evaluations in app store reviews, and how they affect UX?*”. We identified 24 publications and 31 unique factors that could affect UX.

The results from this systematic mapping study revealed a varied effect of these factors. Privacy and Ethical, for instance, had the greatest negative impact on UX, indicating that practitioners should be cautious about what information they collect from users. Other factors, in turn, were more prominent for a given type of app. For instance, Attractiveness, Stability, and Cost were the top 3 factors across the games category, suggesting that practitioners should invest their efforts in these factors to provide better experiences for this particular category of app.

These findings highlight that several factors can influence the experience with varied effects according to the context. However, analyzing thousands of reviews to extract this information is costly and time-consuming. In this scenario, an approach that automatically analyzes the reviews from a given app would be helpful to investigate which factors are leading to positive and negative UX and identify improvement opportunities according to its specificities.

CHAPTER 5 – INVESTIGATING PRACTITIONERS' PERCEPTIONS TOWARDS AN AUTOMATED APPROACH TO ANALYZE APP STORE REVIEWS

This chapter presents an exploratory study and a feasibility study to investigate practitioners' perceptions of our proposal to develop an automated approach that analyzes app store reviews.

5.1. INTRODUCTION

In the previous chapters, we investigated the problem through an empirical study and a systematic mapping of the literature. The results revealed that several factors could affect users' perception of the experience, providing us with the necessary theoretical foundation to develop our proposal and achieve our goal, i.e., support the mobile software development process by proposing an approach that automatically analyzes user reviews from app stores to identify the factors that are affecting UX.

This chapter presents our third iteration over the relevance cycle. In Section 5.2, we present an exploratory study investigating how app store reviews are used by practitioners from the industry, the challenges involved in this process, and their perceptions of an automated analysis approach. The results from this study allowed us to identify the problems faced by practitioners and how an automated approach could support them in their tasks. In Section 5.3, we present a feasibility study we conducted to assess the acceptance of an automated approach from practitioners. To do so, we developed a prototype that extracts the reviews from app stores and presents the most frequent terms in a word cloud. The results from this feasibility study allowed us to identify the practitioners' needs, the limitations of this approach, and improvement opportunities, which served as input to the development of UX-MAPPER.

5.2. EXPLORATORY STUDY

In this section, we present an exploratory study to understand how practitioners analyze user reviews from app stores, their importance in the software development process, and the challenges involved. We also investigated the opinion of practitioners towards an automated approach to analyzing app store reviews.

5.2.1. Participants and Materials

We carried out semi-structured interviews with three practitioners from distinct software development companies in Manaus working on projects developing mobile applications, selected by convenience. The participants had the following profiles: a software tester, a developer, and a project manager. To conduct the interview, we defined five questions related to their main activities, the need for analyzing user reviews, and the main challenges involved in this process as follows:

1. What are the main activities in your work?
2. Have you had to analyze user reviews/feedback to look for improvements or solutions for the company?
3. What is the greatest difficulty in analyzing feedbacks?
4. How could search automation in user-generated texts help your work?
5. What would you expect from this tool?

5.2.2. Procedure

Due to the context of the worldwide COVID-19 pandemic scenario, we conducted the interviews through the Google Meet platform. Before starting the interviews, the participants signed an online consent form, stating that the participants' data would be treated anonymously and we would not publish any sensitive information. Then, we asked the questions presented in the previous subsection. Finally, we thanked the participants and ended the interview.

5.2.3. Results

The results indicated that the three participants had analyzed user reviews to improve the company's software. The software tester, for instance, stated, "*currently, in our project group, it is necessary at a certain time to look at user reviews about our application to see the problems they reported and what we can improve*." It indicates that the companies are aware of the importance of user reviews for software development and evolution.

Regarding the main challenges, two interviewees pointed out the lack of constructive information in the reviews, which hinders the identification of what aspects of the software to improve or fix. The software tester, for instance, stated, "*users sometimes do not make a constructive criticism; they do not tell you what should be improved in the application explicitly*." Two interviewees also pointed out the time required to read and analyze the reviews, making a manual analysis unfeasible. The project manager, for instance, stated, "*[the problem is the] waste of time reading lines and lines of feedback to transform them into a few lines of technical terms*."

Finally, when asked about the helpfulness of an automated analysis of user reviews and their expectations about a tool designed for this task, all three stakeholders were unanimous in saying that it would contribute a lot to their work, especially to speed up the development process. The software tester, for instance, stated, "*this would be very interesting. In addition to showing constructive comments, filtering what we need would be very cool, [i.e., extracting] both negative feedbacks, by which we could improve our application, as well as positive feedbacks. This will greatly automate our work*".

5.2.4. Discussion

This study revealed that users' opinion is essential for the stakeholders to identify potential issues and improvement opportunities to increase users' satisfaction. However, they spend a lot of resources and people to streamline the process of adapting their products to their customers' needs and desires, as there are a huge number of reviews to analyze, most of them not informative. This finding is by Chen et al. (2014), who identified that only 35.1% of the reviews contained information that can directly support developers in improving their software applications. In this sense, the results revealed that the problem under study is relevant, and there is a need to extract meaningful information to support practitioners in their tasks.

We found that automating user reviews' analysis could help them identify the main issues and speed up the development process. It highlights the need for approaches that automate the analysis and provide relevant information for the development team to improve the company's software. Thus, an automated approach can give the stakeholders a more comprehensive view of the application's main issues, which can help them extract and identify what should be prioritized.

5.3. FEASIBILITY STUDY

The exploratory study provided evidence that practitioners consider the feedback from users in app stores valuable. However, the analysis of such reviews is time-consuming and costly to be performed manually, especially considering that many reviews do not provide useful information to improve the app.

This study aimed to answer the following research 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.

5.3.1. Initial Proposal and MVP

Figure 5.1 presents an MVP (Minimum Viable Product) of our initial proposal. First, the user selects the app they want to analyze (see Figure 5.2a). Considering the pandemic scenario, we focused on extracting reviews from technologies that support remote teaching. Among them, we selected Kahoot!, one of the most popular game-based learning platforms, with over 2.5 billion people from more than 200 countries since its release in 2013 (Wang and Tahir, 2020), and used by 87% of the top 500 universities around the world[6]. After selecting the application, the tool gets the reviews from this app, ordered by date. For this task, we adopted the Google-Play-Scraper API for Python[7], which allows extracting user reviews from Google Play Store. After getting the reviews, the tool performs a preprocessing step by expanding contractions and removing stopwords, HTML tags, URLs, e-mails, and accented characters. It also applies lemmatization, which reduces different inflected forms of a word into their lemma for being analyzed as a single word (Maalej and Nabil, 2015). This grouping is essential to reduce the number of feature descriptors to be analyzed when extracting the most frequent features of the app under analysis. By applying lemmatization, words such as 'crash', 'crashing', and 'crashes' are grouped into the term 'crash', which will increase its count and help to identify the most frequent terms.

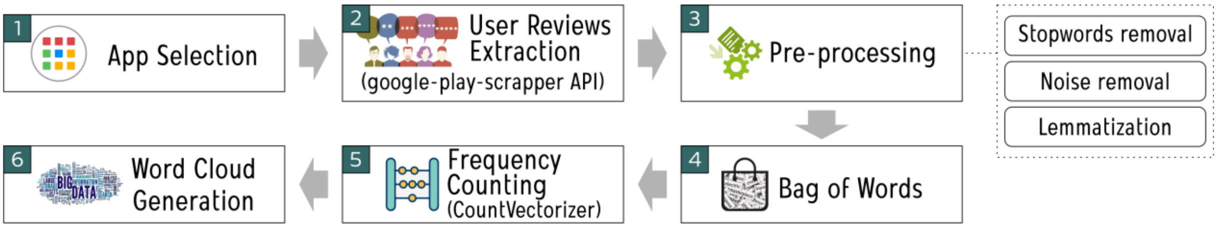


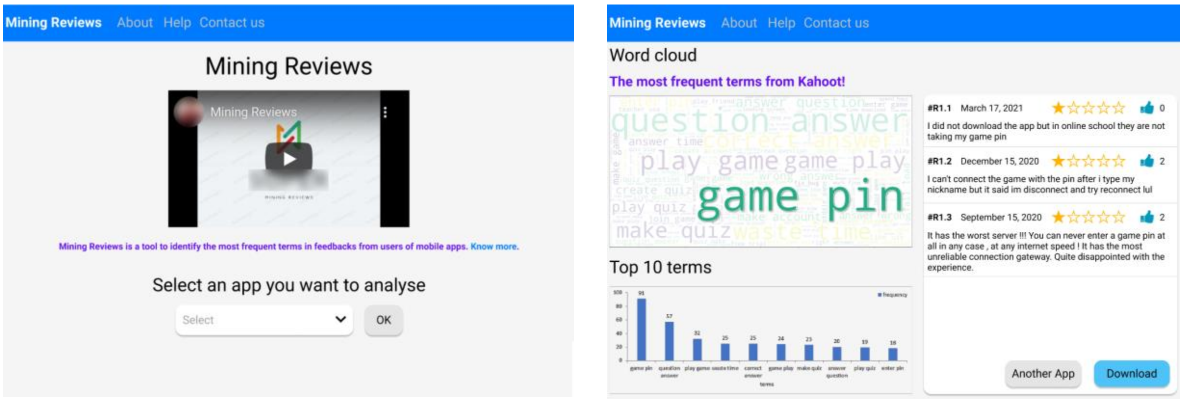
Figure 5.1 - Overview of our initial proposal.

For this task, we adopted SpaCy, a state of art natural language processing tool (Al Omran and Treude, 2017). We selected lemmatization over stemming because the latter may lead to words that are not understandable, for instance, by transforming 'confusing' into

[6] https://techcrunch.com/2020/06/11/kahoot-raises-28m-for-its-user-generated-educational-gaming-platform-now-valued-at-1-4b/

[7] https://pypi.org/project/google-play-scraper/

‘confus’. It can also lead to misleading words, such as transforming ‘care’ into ‘car’. Moreover, stemming does not consider the context of the term (Nayebi et al., 2018), making it unable to distinguish between words with different meanings. For instance, while lemmatization recognizes ‘added’ as the lemma of ‘add’, stemming reduces it to ‘ad’, which may be confounded by the abbreviation of ‘advertisement’.



Mining Reviews is a tool to identify the most frequent terms in feedbacks from users of mobile apps. Know more.

Select an app you want to analyse

The most frequent terms from Kahoot!

I did not download the app but in online school they are not taking my game pin

I can't connect the game with the pin after i type my nickname but it said im disconnect and try reconnect lul

It has the worst server !!! You can never enter a game pin at all in any case , at any internet speed ! It has the most unreliable connection gateway. Quite disappointed with the experience.

(a) (b)

Figure 5.2 – Screenshots of the Mining Reviews prototype.

In the next step, the tool creates a Bag of Words (BoW), a dictionary with all the terms extracted from the reviews and their frequency in the corpus. From this BoW, the tool extracts bi-grams and tri-grams, i.e., all combinations of two or three contiguous words in a review (Maalej et al., 2016). We decided not to extract single words as they cannot convey the context in which it is inserted. For instance, the word ‘fails’ indicates that the application may have some problems. However, bi-grams and tri-grams such as ‘never fails’, ‘always fails’, and ‘fails during startup’ have very different meanings.

Finally, the tool generates a word cloud of bi-grams and tri-grams. The terms are presented in a variety of colors and different sizes according to their frequency. When the practitioner clicks on a term (see Figure 5.2b), the tool shows a list of reviews that contain this term. Each review comprises the reviewer's name, title, date, star rating, and the number of likes. The tool presents the reviews ordered by date and the number of likes they received.

5.3.2. Participants and Materials

We carried out this study with six practitioners from distinct companies that did not participate in the exploratory study (see Table 5.1). Due to the difficulty of recruiting practitioners from the industry, the selection of the participants was by convenience. We selected the participants considering the following criteria: (i) participants working on software development companies and (ii) participants with prior experience in requirements elicitation. 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.

Table 5.1 - Participants' profile.

	Role	Exp. (years)	Exp. in req. elicitation	Exp. in req. elicitation through reviews
P1	Quality assurance analyst	1-3	Moderate	Moderate
P2	Req. engineer / P.O.	10+	High	None
P3	Req. engineer / P.O.	10+	High	None
P4	Developer	4-6	Moderate	Low
P5	Quality assurance analyst	1-3	Low	None
P6	Developer, req. engineer, tester, project manager	4-6	Moderate	None

We used the following materials in this study: i) an informed consent form; ii) a characterization questionnaire to gather participants' information on their role in the development team, years of experience in this role, experience in requirements elicitation, and experience in eliciting requirements from user reviews; iii) the Mining Reviews tool; iv) a spreadsheet to report the elicited requirements; v) a post-study questionnaire comprising the Technology Acceptance Model (TAM) (Davis and Venkatesh, 1996) with additional open-ended questions to obtain participants' opinion on what was easy or difficult when using the tool and improvement suggestions. The TAM consists of a 7-point Likert questionnaire designed to assess the acceptance of technology through various constructs, such as Computer Self-Efficacy, Perceived Enjoyment, Objective Usability, and others. Among them, we focused on the three core constructs: Perceived Usefulness (PU), Perceived Ease Of Use (PEOU), and Behavioral Intention (BI). We adapted this questionnaire to fit into our context by changing some keywords of each statement, as presented in Table 5.2.

Table 5.2 - Items from TAM questionnaire adapted for this study.

Item	Statement
PU1	Using the tool improves my performance in extracting requirements through app store reviews
PU2	Using the tool improves my productivity in extracting requirements through app store reviews
PU3	Using the tool allows me to fully extract requirements through app store reviews
PU4	I find the tool useful for extracting requirements through app store reviews
PEOU1	The tool was clear and understandable
PEOU2	Using the tool did not require a lot of my mental effort
PEOU3	I find the tool to be easy to use
PEOU4	I find it easy to think about requirements through app store reviews using the tool
BI1	Assuming I had access to the tool, I intend to use it
BI2	Given that I had access to the tool, I predict that I would use it
BI3	I plan to use the tool in the next months

5.3.3. Procedure

Two researchers conducted the study with three participants each. Due to the COVID-19 restrictions, we carried out the study entirely online through the Google Meet platform[8]. First, we introduced the study and its goals. Then, we asked the participants to sign in the informed consent form and the characterization questionnaire. Next, we sent them the link to the tool for the participants to explore for some minutes and explained its functionalities. Then, we asked the participants to think about requirements to improve the app or develop a concurrent application based on the reviews presented in each of the five most frequent terms. Finally, we asked them to fill in the post-study questionnaire.

5.3.4. Results

This section presents the results from the requirements elicitation process, the TAM questionnaire, and the open-ended questions.

5.3.4.1. Requirements Elicitation

The participants spent, on average, 37min10s to interact with the tool and think of requirements based on the reviews (standard deviation of 16min03s). Participant P6 required the least amount of time (16min), while participant P4 required the most (1h03min).

To analyze the results better, one researcher read the requirements and merged those with similar descriptions, and another researcher reviewed the process. We excluded descriptions that only presented the opinion of the participant and did not highlight any requirement. Disagreements between the researchers were solved through a discussion session. From the 35 requirements reported, 26 were unique (see Table 5.3). It is noteworthy that the participants did not perform the requirements specification process. Thus, the output is a list with a set of candidate requirements.

Table 5.3 - Number of requirements elicited and time spent by participant.

	P1	**P2**	**P3**	**P4**	**P5**	**P6**
Req.	4	9	4	3	11	4
Unique	3	8	2	2	8	3
Time	00:27:35	00:39:30	00:43:35	01:03:20	00:32:58	00:16:00

Overall, the participants were able to think about requirements based on the reviews associated with the top 5 terms presented by the tool. Surprisingly, participant P5 reported the greatest

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

number of requirements (11), even having low experience in requirements elicitation and no experience analyzing user reviews for this goal. Such a result indicates that our approach can support practitioners regardless of their experience.

It is noteworthy, however, that not all reviews led to requirements (see Table 5.4 to visualize all the reviews included in the prototype). The number of requirements obtained from each term were as follows: game pin (7), question answer (6), correct answer (5), play game (5), and waste time (3). Reviews associated with the term ``waste time" had the lowest number of requirements elicited. The participants did not elicit requirements from two out of three reviews (R4.1 and R4.3). Such reviews did not provide sufficient information for the practitioners to think about requirements, as they were purely emotional.

Table 5.4 - Reviews sample included in the MVP.

Terms	ID	Review
game pin	R1.1	I did not download the app but in online school they are not taking my game pin
	R1.2	I can't connect the game with the pin after i type my nickname but it said im disconnect and try reconnect lul
	R1.3	It has the worst server !!! You can never enter a game pin at all in any case , at any internet speed ! It has the most unreliable connection gateway. Quite disappointed with the experience.
question answer	R2.1	Some of the questions answer are biblically wrong
	R2.2	Cant see the question or answers just the shape's so when playing with others I've got to guess what colour/ shape rather than reading the question and choosing the answer
	R2.3	This app ***** the questions and answers should be readed to be understandble
play game	R3.1	Why do we have to log in to play? To play your own game
	R3.2	Questions aren't visible while playing the game with my friends nor the options!
	R3.3	I was happy with this app until it started crashing. I go to play games and I just hear corny music. The questions don't come up.
waste time	R4.1	I have never experienced any app in the world to be such a rubbish. Tihs app is just waste of time. Really worst app in the world.
	R4.2	Cannot see questions on device. Lag in connectivity affects video meetings and prevents timely answering. Waste of time.
	R4.3	I hate this app and can not even use it. WASTE OF TIME.
correct answer	R5.1	sometimes it lags and wont let me press the correct answer
	R5.2	I have play kahpot in 1 year.It good app but something wrong after multi - select. I choose 3 of 4 and answer partially correct.But all answers I choose I correct. This made me angry, fall ranking challenge from 6th to 18th. Please fix this
	R5.3	I hit the correct answer but it said times up

5.3.4.2. TAM Questionnaire

Overall, most of the participants agreed with all the statements from TAM. The median for Perceived Usefulness (PU), Perceived Ease Of Use (PEOU), and Behavioral Intention (BI) were 6, 6, and 6.5 respectively, indicating a positive perception of the participants regarding our proposal (see Figure 5.3).

Regarding PU, most of the participants considered our approach useful. The PU2 and PU4 items were those with the highest level of agreement among the participants. They considered that the tool improves their performance to explore requirements from user reviews (PU2) and is, in general, useful for supporting this task (PU4). However, participants P2 and P4 disagreed that the proposed approach completely allows thinking about requirements through app store reviews (PU3), indicating that the reviews did not provide enough information for this task.

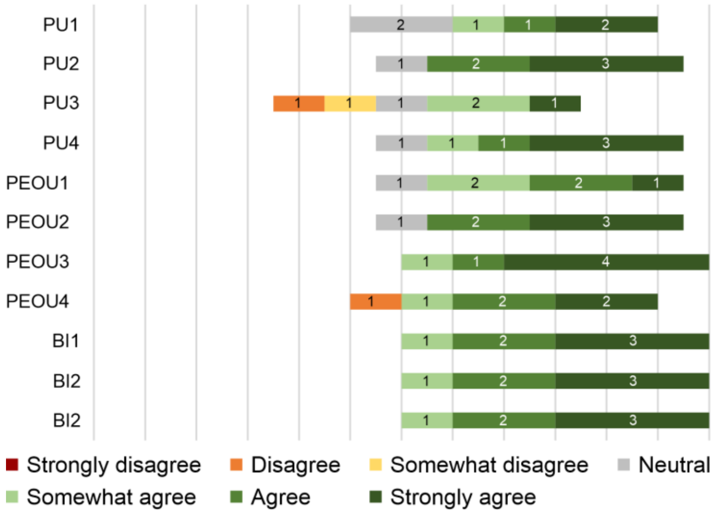


Figure 5.3 - Results from the TAM questionnaire.

Regarding PEOU, the opinions were slightly more positive. Most of the participants agreed with all the fours statements, especially the PEOU2 and PEOU3 items. They considered that the tool did not require too much mental effort (PEOU2) and was easy to use (PEOU3). Participant P2, however, did not agree with PEOU4 ("*I find it easy to elicit requirements through app store reviews using the tool*"). It reinforces that this participant faced difficulties in the requirements elicitation process.

Finally, regarding BI, the results indicate a positive acceptance of our proposal from the practitioners. They were unanimous in affirming that they would use the tool.

5.3.4.3. Open-ended questions

When asked whether they faced any difficulty in thinking about requirements from the reviews, all the participants were unanimous in affirming that, in general, they did not face any problem in this task. It explains the high level of agreement with the PEOU items from TAM. However, some participants pointed out some issues. Participants P2, P4, and P5 reported the need to understand the context to analyze the reviews to think about requirements. Participant P2 stated "*some reviews lack context and seem like general responses. It was necessary to understand the context of what was happening to the user to extract the requirement."* Participant P4 also stated "*it requires a lot of interpretation to carry out the analysis of these comments."* These quotations might explain the disagreement of these two participants in the TAM questionnaire when asked about the usefulness of the approach to elicit requirements from user reviews fully (PU3). Finally, participant P5 reported that "*in 3 cases, the reviews passed by the tool did not have a requirement (functional or non-functional) that could be extracted from it. For extracting a requirement, context and situation of the problem should be contained in the comment.*"

Regarding what was easy when using the tool, the participants mentioned the visual representation through the word cloud and the top issues through the bar plot. They considered that they were easy to interact with and helped to identify the most important issues. Participant P1 stated "*it was easy to] identify the important points from users' perception through the top issues. It makes it easier for those who are doing this work.*" Participant P5 reported "*[it was easy to] quickly and visually extract the most mentioned points and most important words through the word cloud.*" Finally, participant P6 stated "*it helps on identifying requirements, as it presents the most frequent terms and makes it easy to get an overview of all reviews made by users.*"

Regarding what was difficult when using the tool, participants P1 and P6 did not report any issue. For the other participants, the difficulties vary. Participant P2 had difficulty identifying if the "thumbs up" count was for the review or the star rating. Participant P3 considered it difficult to understand the reviews without knowing the app under evaluation. Participant P4 considered the word cloud a little confusing initially, but after understanding how it works, he found it very practical and straightforward. Finally, participant P5 stated "*[it was difficult] to identify the most cited words shown in the [top 10 words] graph into the word cloud.*"

When asked what they would change to improve the tool, the participants provided various suggestions. Participants P1 and P4 suggested improving the interface to make it look more professional (P1) and more intuitive when selecting the terms in the word cloud (P4). Participants P2, P3, and P5, in turn, suggested new functionalities. They asked to add options to sort the reviews by the number of likes and ratings (P2), to search for a specific term in the reviews (P3), and to be able to click on the words shown in the top 10 words graph instead of searching and clicking on the word cloud. Finally, participant P6 suggested using another graphical representation, as the word cloud does not show the frequency of the terms, and it is hard to compare them visually.

5.3.5. Discussion

This study highlighted the potential of an automated approach to support practitioners in identifying the main issues from app store reviews. Despite the difficulties some participants faced when thinking about requirements from user reviews, all of them were unanimous in affirming that they would use the tool if it were made available, indicating its positive acceptance. The visual representation through a word cloud and the ranking into the top 10 frequent terms made it easy for practitioners to identify the main issues in the application quickly. Moreover, the reviews associated with these terms supported practitioners to think about requirements.

Regarding the limitations of this approach, some participants pointed out that some reviews require much interpretation to analyze due to the lack of context. It is probably because most of the participants did not know how Kahoot! works. In a scenario where the practitioner analyzes reviews from their app, it may be easier to identify potential problems and understand what the users are talking about. However, when it comes to analyzing reviews from competing apps, the lack of understanding of the app may make it difficult to think about requirements. In such a scenario, practitioners should explore the competing app and its functionalities to understand how it works before analyzing the reviews.

Another problem raised by a participant is that not all reviews led to requirements, especially those containing the ``waste time" term. We identified that two out of the three reviews were purely emotional, not providing any information that would lead to a requirement. It is by Palomba et al. (2018), who stated that reviews with pure emotional expressions do not present any relevant information to be linked to an issue. According to Chen et al. (2014), only

35.1% of the reviews contain information that can directly help developers to improve their apps, highlighting the challenge of filtering such reviews.

Regarding improvement suggestions, the participants highlighted important features to be added to the tool. The request to include filtering and sorting functionalities indicates that practitioners need more flexibility to analyze the reviews. The request for a representation that allows comparing the frequency of each term also raises the need for an approach that presents this information more precisely to support the decision on which issues to focus on better.

5.4. SUMMARY

This chapter presented an initial proposal of an approach that automatically analyzes user reviews from app stores. To evaluate the feasibility of this approach, we conducted a study with six practitioners from the industry with experience in requirements elicitation.

Overall, the results indicated a positive acceptance of our proposal. The participants found it easy to use, understand, and overview the main issues through a comprehensive view using word clouds. However, we identified points that need to be improved: i) the presence of some reviews not relevant for requirements elicitation; ii) the need for filtering and sorting functionalities; iii) an alternative approach that allows comparing the frequency of each term more precisely.

Regarding the first issue, we could implement more specialized feature extraction approaches instead of presenting the most frequent terms without any criteria, improving the extraction of more relevant terms. Regarding the second issue, we could classify the reviews according to the factors we identified in the literature, making it easier for practitioners to find reviews related to specific topics, such as bugs and improvement suggestions. The tool could also have functionalities to sort and filter the reviews by the number of thumbs up and star ratings. Previous works have demonstrated that users consider long and detailed reviews more helpful (Palomba et al., 2018; Simmons and Hoon, 2016), making this metric a good alternative to identifying useful reviews that convey the opinion of different users. Finally, regarding the third issue, we could use a bar graph to present the top features, making it easier to compare their frequencies.

CHAPTER 6 – UX-MAPPER DEVELOPMENT PROCESS

This chapter presents the development of UX-MAPPER. We detail its architecture, the steps performed to develop and refine our model, and an example of its functioning.

6.1. INTRODUCTION

In the previous chapters, we investigated the problem by conducting exploratory and feasibility studies. Overall, the empirical studies, as well as the systematic mapping study, provided the theoretical foundation needed to develop our artifact. We present how we filled the gaps and applied the knowledge obtained to develop UX-MAPPER for each finding from the previous studies.

Findings from the first empirical study (CHAPTER 3):

- **Different factors can affect UX evaluations:** we found that some factors affect users' perception of their experiences. The type of method employed (inspection or testing) and previous experience with similar products significantly affected users' evaluations. On the other hand, the interaction sequencing factor, pointed out in the literature as having strong influence on users' evaluations in controlled settings may not have a significant effect during actual usage with real applications, raising the need to investigate the impact of different factors on UX.
    - Such findings led us to perform a systematic mapping study to investigate what is known in the literature regarding influencing factors. From this study, we identified 31 factors, which served as the basis for developing UX-MAPPER.
- **Effect of previous experience on ratings:** this finding highlights the importance of considering similar software applications when designing a new app or when improving an existing one. As users can make their previous experiences a baseline, it is important that software developers know the apps from their competitors to identify their most liked features, the most hated ones, and improvement requests.
    - Taking this finding into account, UX-MAPPER should allow practitioners to select the competing app and investigate what factors and features are leading to positive or negative UX. Such information could be useful for practitioners to define which factors and features need to be prioritized.

Finding from the systematic mapping study (CHAPTER 4):

- **Impact of the factors unclear:** the systematic mapping study allowed us to obtain a set of factors and their associated polarity (positive, negative, or neutral). However, their impact on users' evaluations and sentiments remains unclear, which makes it difficult for developers to identify which factor to prioritize in the development process.
    - This finding indicates that UX-MAPPER should allow the practitioner to classify the reviews of a given factor by relevance (i.e., the number of thumbs up received by other users), which would allow identifying features that several users are requesting to fix or to be implemented. Moreover, the possibility of filtering the reviews by the star ratings would make it possible to analyze the distribution of the reviews for each factor and feature to verify the ratio between positive and negative ratings to assess their impact.
- **Factors with different effects according to the context:** some factors can have varied effects. A positive perception of usability, for instance, does not result in better evaluations, but a negative perception can affect evaluations significantly more negatively. Other factors are considered more critical for a given type of app. For mobile games, for instance, Attractiveness, Stability, and Cost were the top 3 factors for the entire category.
    - It highlights the importance to develop an approach that automatically analyzes user reviews and allows identifying which factors and features to prioritize for each app or group of apps.

Findings from the exploratory and feasibility studies (CHAPTER 5):

- **Practitioners take user feedback from app stores into consideration during software development and evolution:** the interviews we conducted with practitioners indicate that they need to analyze app store reviews to obtain user feedback regarding problems and improvement opportunities at a given moment of the software development process.
    - This finding strengthens the importance of our proposal for software development and evolution.
- **It is time-consuming to identify relevant reviews:** despite the benefits of analyzing user reviews, identifying reviews that provide constructive information that can lead to improvements or fixes is time-consuming to be performed manually. Practitioners said that filtering these reviews to identify relevant ones that provide

positive and negative feedback on what they should fix and improve would greatly help their work.

  - This finding indicates that UX-MAPPER should classify and filter the reviews into factors to facilitate finding reviews related to a given topic, such as bugs, improvements, and performance issues. It should also extract features from the reviews to present the most relevant ones for practitioners.

- **WordClouds are not the best way to present features:** some participants reported that they would want to compare different features more precisely, which is hard to do in a word cloud with words with very similar sizes.
  - UX-MAPPER should provide a graphical representation that makes the data comparison and analysis more intuitive, such as a bar graph ordered by the frequency of each feature to create a rank that allows comparing the features more precisely.

In this chapter, we present our first iteration over the Design Cycle, which consists on developing and evaluating the artifact. The following subsections describe the UX-MAPPER architecture, its development and refinement process, and its functioning.

6.2. UX-MAPPER ARCHITECTURE OVERVIEW

In this subsection, we present an overview of the UX-MAPPER Architecture. It is organized into three main components (see Figure 6.1): 1) Data Gathering and Processing Component; 2) Factor Extraction Component; and 3) Feature Extraction Component.

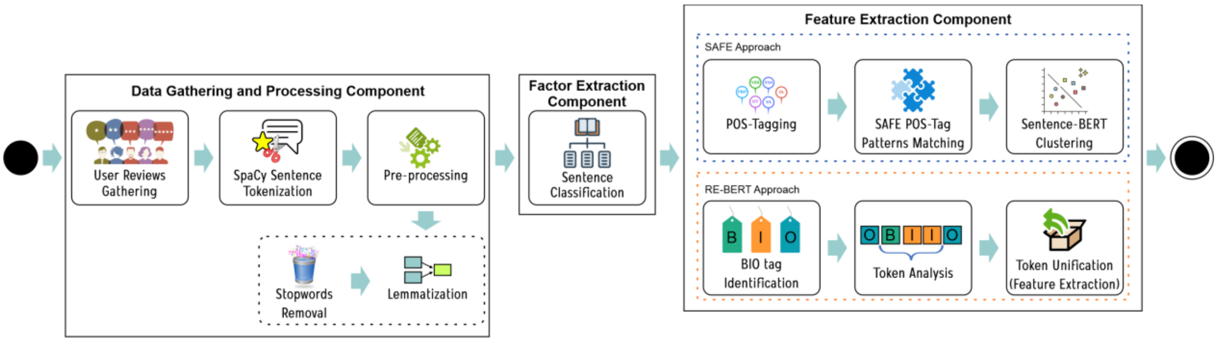


Figure 6.1 - UX-MAPPER Architecture.

The **Data Gathering and Processing Component** is responsible for obtaining user reviews from app stores and performing text processing to prepare the data that will serve as input for the other components. It uses the Google-Play-Scrapper API for Python to extract the reviews, which allows obtaining information for both app (e.g., version, number of downloads, recent changes) and user reviews (e.g., content, username, rating) in a JSON format, making it easy to work with. After obtaining the reviews, it performs sentence tokenization to split the reviews into sentences using SpaCy, a state-of-the-art natural language processing tool (Al Omran and Treude, 2017). Then, the component performs some preprocessing steps. First, it cleans the data by removing stopwords, such as articles, pronouns, prepositions, and conjunction. Then, it uses SpaCy to reduce different inflected forms of a word into their lemma.

The **Factor Extraction Component**, in turn, takes the output of the Data Gathering and Processing Component to analyze the data and tag the sentences according to the factor identified. Each sentence can be classified into more than one factor. For this task, the component uses Support Vector Machine (SVM), a supervised classifier that have been proved to be highly effective on a variety of tasks, such as text classification, pattern recognition, and computer vision (Nalepa and Kawulok, 2019). To make the classifier recognize the factors in each sentence, we trained and fine-tuned it iteratively. Details of its implementation are presented in Subsection 6.3.

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. In this component, we implemented two different 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). The former employs a pattern-based extraction by analyzing the Part-Of-Speech (POS) Tags to extract relevant features, while the latter uses a machine learning approach. We employed these two approaches to investigate the usefulness and relevance of their outcomes from practitioners' perspective (see Section 7.4). Details of their implementation are provided in Section 6.4.

To develop these components, we followed a set of steps. Figure 6.2 presents an overview of the UX-MAPPER development workflow. In the next subsections, we detail each of these steps.

6.3. FACTOR EXTRACTION COMPONENT

To develop this component, we followed a set of steps. First, we selected the factors we identified in the systematic mapping study, according to their applicability to our context (Step 1). Then, we obtained a sample of user reviews from five different app categories (entertainment, communication, tool, social, and game) and tokenized them into sentences for labeling (Step 2). Next, we conducted a pilot study before labeling a large dataset. The goal was to assess the level of agreement between the researchers, discuss the disagreements, refine the definition of the factors, and perform new labeling if necessary (Step 3). Then, we performed a set of iterations to evaluate different classifiers and select the most suitable to be employed in UX-MAPPER (Step 4).

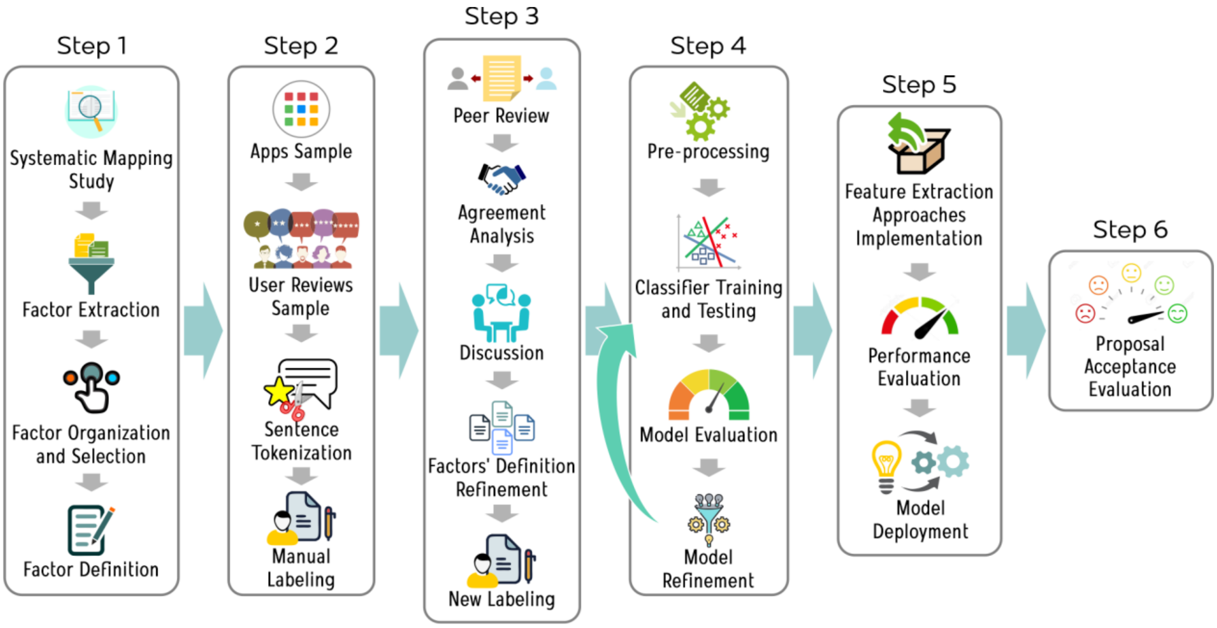


Figure 6.2 - UX-MAPPER development workflow.

6.3.1. Factors Organization and Selection

The first step to developing our method was organizing and selecting the factors returned in our systematic mapping study. First, we analyzed each factor and checked whether it can be addressed through user reviews and ratings. We removed “Findbugs Warnings” and “Presence of test cases” because both rely on source code analysis, which is out of the scope of this research. We also removed “Device model” and “Culture”, as they are not publicly available in the reviews, except for the owner of the app itself, and “Gender”, as it did not have a significant influence on users’ evaluations. We also did not consider the “User Profile of an App Type” factor, as it requires a comparison of an entire app category, which can consume a lot of resources and processing time. Although “App version” and “Date/Time” also did not affect users’ judgments, we did not remove these factors. It is because the publication reporting the findings obtained low scores in our quality assessment analysis. Thus, we cannot conclude that these factors, in fact, do not affect UX evaluations. Due to this, we decided to keep these factors in the method for further analysis. We also did not consider “Feature/Functionality” as they can vary according to each app and are also mentioned as part of the problem/suggestion in reviews from other factors, such as “Improvement request”, “Bugs/Crash”, and “Feature removal”, being possible to identify them, for instance, through collocation algorithms. Finally, due to overlapping issues identified during the pilot study (see Section 6.3.3), we merged “Network Problem” into “Bugs/Crash” factor. We also merged “Ease of use” and “Simplicity” into “Usability”, as both are related to the pragmatic aspects of the experience, i.e., usability. Table 6.1 presents the list of factors we adopted and removed/merged from UX-MAPPER.

Table 6.1 - List of factors considered and removed/merged from UX-MAPPER.

Factors considered in UX-MAPPER			**Factors removed/merged**
Accuracy	Feature removal	Resource use	Culture
App version	Helpfulness	Spam/Ads	Device model
Attractiveness	Improvement request	Update	Ease of use
Bugs/Crash	Interface	Usability	Feature/Functionality
Comparison	Misleading app		FindBugs warnings
Compatibility	Performance		Gender
Cost	Personalization		Network problem
Customer support	Privacy and Ethical		Presence of test cases
Date/Time	Recommendation		User profile of an app type

6.3.2. Evaluation Metrics

According to Zhang and Zhou (2014), there are a variety of metrics proposed to evaluate multi-label learning, which can be categorized into two groups: example-based and label-based metrics. The *example-based* metrics consist of assessing the model's performance on each test example separately. Then, the mean value across the test set is returned. The *label-based* metrics, in turn, calculate the performance on each class label separately, and then return the macro/micro-averaged value across all class labels. As we have multiple labels and want to evaluate their performance individually, we adopted the *label-based* metrics.

There are four basic quantities needed to characterize the performance of the binary classification per label: True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN). Let x_i be the i-th instance (sentence) from the labeled dataset x, Y_i the set of labels associated to instance x_i, and y_j the j-th label predicted by the classifier for this instance through the function $h(x_i)$, we can represent these basic quantities as follows (M.-L. Zhang and Zhou, 2014):

$$TP_j = |\{x_i | y_j \in Y_i \wedge y_j \in h(x_i), 1 \leq i \leq p\}|$$

$$FP_j = |\{x_i | y_j \notin Y_i \wedge y_j \in h(x_i), 1 \leq i \leq p\}|$$

$$TN_j = |\{x_i | y_j \notin Y_i \wedge y_j \notin h(x_i), 1 \leq i \leq p\}|$$

$$FN_j = |\{x_i | y_j \in Y_i \wedge y_j \notin h(x_i), 1 \leq i \leq p\}|$$

Based on these quantities, we can calculate the Precision, Recall, and F1-score binary classification metrics to measure the performance of the classifiers. *Precision* refers to the number of correct predictions (TP) divided by the number of all predictions made by the classifier (TP + FP). Recall, in turn, is the ratio between the number of correct predictions (TP) and all observations in actual class, i.e., the sum of the instances predicted and missed by the classifier for a given class (TP + FN). Finally, the *F1-score* gives a metric for the balance between Precision and Recall. These metrics are represented by the formulas below:

$$Precision_j = \frac{TP_j}{TP_j + FP_j}$$

$$Recall_j = \frac{TP_j}{TP_j + FN_j}$$

$$F1_j = 2 * \frac{Precision_j * Recall_j}{Precision_j + Recall_j}$$

Finally, we can calculate the *macro* and *micro* label-based metrics for each binary classification metrics, i.e., precision, recall, and f1-measure. The *macro-averaging* is the sum of the result of the binary classification metric from all classes divided by the number of classes. The *micro-averaging*, in turn, aggregates the basic four quantities (TP, FP, TN, and FN) to be treated as a unique metric to calculate each of the binary classification metrics. Considering *B* as the target binary classification metric, and *q* as the number of classes, we can represent the label-based metrics as follows (M.-L. Zhang and Zhou, 2014):

$$B_{macro}(h) = \frac{1}{q}\sum_{j=1}^{q} B(TP_j, FP_j, TN_j, FN_j)$$

$$B_{micro}(h) = B(\sum_{j=1}^{q} TP_j, \sum_{j=1}^{q} FP_j, \sum_{j=1}^{q} TN_j, \sum_{j=1}^{q} FN_j)$$

6.3.3. Pilot Study

Before labeling a large set of reviews, we first performed a pilot study. To diversify our sample and cover a variety of reviews, 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. Each review consists of one or more sentences and can contain one or more factors. To facilitate the classification process, we tokenized the reviews at sentence level of granularity, which also allows the ML model to learn more effectively as the number of words that can be associated to a given factor is reduced. To do so, we applied the sentence tokenizer from the NLTK library, which is widely used in many works in the field (Bakiu and Guzman, 2017; Harman et al., 2012; Hedegaard and Simonsen, 2013; Palomba et al., 2017). The segmentation resulted in 51 sentences.

This pilot study involved four people: the main researcher of this work and three computer science undergraduate students. These students were involved in a research project related to the extraction of software requirements from user reviews. Prior to the labeling process, the main researcher presented the context of the research to the students, and explained about each factor by providing their definition and some examples. The students also performed a labeling exercise and had their questions answered by the researcher. Each person performed the labeling process individually. Additionally, the three students had to discuss their classifications together and reach a consensus to provide a single labeled set.

Six out of 51 sentences had disagreements. The main cause of the disagreements was due to the Bugs/Crash and Network Problem factors. We identified that it may be difficult to identify whether the problem reported by the user is due to connectivity problems or some bug in the app. Consider, for instance, the following sentences: "*Can't open app keeps telling me it can't connect to Netflix service*", and "*Tiktoks videos won't load again*". It is hard to guess the cause of these problems. After discussing the disagreements, we decided to merge the Connectivity factor into the Bugs/Crash factor, as the definition of the later is broader.

6.3.4. First Iteration

The pilot study allowed us to adjust the factors before labeling a large dataset. This subsection describes the first iteration to build our training set and train and test the classifiers.

6.3.4.1. Tokenization and Manual Labeling

After the pilot study, we followed to the labeling process. During the process, we realized that reviews from the Game category have specificities that would make it hard for the model to learn. Users report many specific problems related to a given stage/phase of the game with a variety of narratives that does not use common words that indicates a bug or a problem, making it difficult to identify a pattern. For example, a user wrote the following review: "*I lost my items in the chest, the orangery flower didn't go up to 1/8 BUT my rainbow blast is lost one and the level that I just passed a while ago just remained*". Another user also wrote "*In new event I'm on scare snake location every time I hits energy but answer that passage block*". 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 progress of the gamer and how the story evolves over time. 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 to any of the factors, giving a total of 666 sentences labeled with at least one factor. Figure 6.3 presents the distribution of instances by factor.

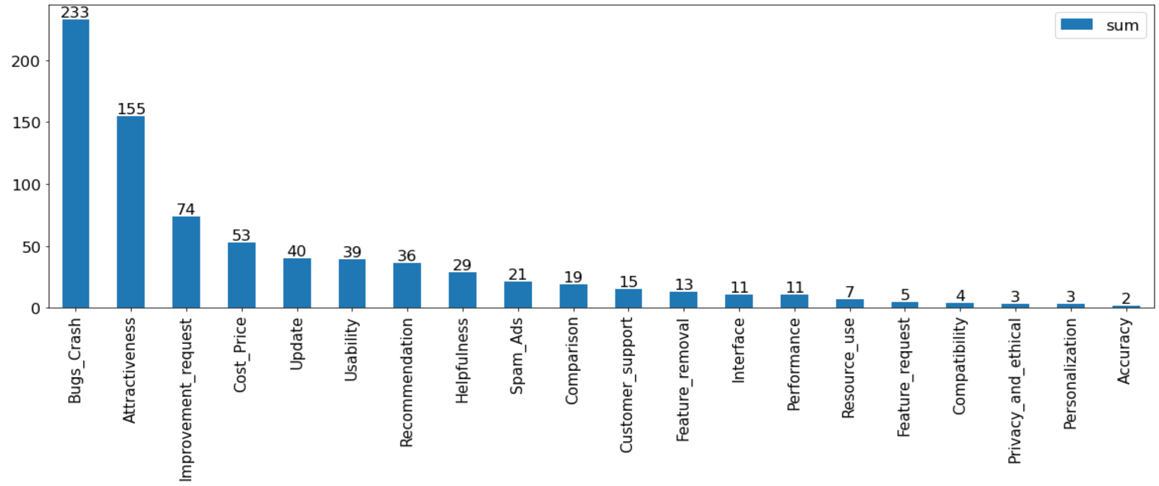


Figure 6.3 - Distribution of labeled instances by factor (first iteration).

6.3.4.2. Preprocessing

In the preprocessing step, we first cleaned up the data. We made the text lowercase, removed text in square brackets, and words containing numbers. Then, we tested with both stemming and lemmatization algorithms to reduce the words into a common base or root, which will result in a smaller number of feature descriptors that the model will need to analyze and learn.

We decided to use lemmatization because it considers the linguistic context of the term and uses dictionaries, while stemmers operates on single words and therefore cannot distinguish between words that have different meanings depending on part of speech (Maalej et al., 2016). During our analysis, we identified that stemming can over-stem, reducing words with different stems to the same root. For example, the words “add, added, adding, ads” are all reduced to “ad”, although the last word refers to advertisements.

We also filtered a set of common terms called “stop words”, such as prepositions, determiners, and conjunctions, to reduce the number of feature descriptors. To do so, we used the set of stop words provided by the NLTK library.

6.3.4.3. Model Training and Testing

We trained our model by employing four classifiers: J48, Logistic Regression, Linear SVC (SVM), and Multinominal Naïve-Bayes. We selected these classifiers as they are commonly applied in the field of user reviews mining, providing good results (Bakiu and Guzman, 2017; Gomez et al., 2015; Hedegaard and Simonsen, 2014; Lu and Liang, 2017; McIlroy et al., 2016; Panichella et al., 2015). Given that the classifiers we selected are of binary type and our problem is of multilabel type (i.e., each sentence can be assigned to more than one label), we used the OneVsRestClassifier algorithm[9] from scikit-learn to make the training and testing process possible.

To create our training set, we first transformed each class (i.e., factor) into dummy-coded variables (e.g. 0- false, 1- true) by using the MultiLabelBinarizer function from the scikit-learn library. This function converts the data into a binary matrix of "samples X classes" that indicates the presence of a class label in each sample (i.e., sentence). After transforming to a binary matrix, we extracted features with one and two words (n-grams = 1,2) from the set of sentences by using CountVectorizer function, which converts text documents into a matrix of token counts. Additionally, we tested with an alternative terms count called TF-IDF (Term Frequency-Inverse Document Frequency). Instead of just counting the frequency, it combines the frequency of the term with the inverse document frequency to calculate the importance of the term in the document (Maalej et al., 2016). In other words, it gives greater weight proportionally to the number of times it appears, but penalizes when it occurs in many or each document. Finally, we divided the labeled dataset into training and testing sets. All the processing was performed in a notebook equipped with Intel Core i7-8565U processor, 8GB DDR4, NVIDIA GeForce MX110 2GB DDR5, Corsair SSD MP510 480GB.

To minimize the bias of random sampling of the training set, we performed a *k*-fold cross-validation. In this approach, the data is split into *k* equal groups (folds), where one of them is selected as the testing set and the remaining ones as the training set (McIlroy et al., 2016). All this process is repeated *k* times with different sets selected as training and testing in each iteration, also known as cross-validation. The cross-validation estimate is a simple average of the *k* individual performance measures as the formula below, where PM is the performance measure of each fold (Oztekin et al., 2013):

[9] https://scikit-learn.org/stable/modules/generated/sklearn.multiclass.OneVsRestClassifier.html

$$CV = \frac{1}{k}\sum_{i=1}^{k} PM_i$$

As our dataset comprises 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, aiming to reduce the possibility of obtaining folds without positive examples, which could affect the results of the classifier. In this study, we performed a 10-fold cross-validation.

6.3.4.4. Results of the First Iteration

Table 6.2 presents the results for each classifier according to the feature extractor algorithm. Regarding *micro-averaged* metrics, the SVM, LR, and NB achieved high precision with over 87% of the instances classified correctly. However, their recall was very low, indicating that they are missing many of the instances. J48, in turn, had poor precision, with around 60% of the instances classified correctly, but achieved greater recall. When it comes to the *macro-averaged* measures, all four classifiers had very poor performance in all metrics. 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. It also had very few effects in classification, with a slightly increase in precision at the expensive of recall, which resulted in a decrease in F1-score.

Table 6.2 – Results of each classifier with k-fold = 10 (first iteration).

	Micro			**Macro**			**Fit Time**
Classifier	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**	**(s)**
SVM (TF-IDF)	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 (TF-IDF)	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 (TF-IDF)	**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 (TF-IDF)	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

To further understand the low performance of the classifiers, we discussed about the labeling process. One issue identified in this first iteration is related to the nature of the reviews. Differently from reviews written in online stores, user reviews from app stores are shorter, given that many of them are submitted from mobile devices on which typing is not easy (Fu et al., 2013). Due to this, they are also usually unstructured, without proper punctuation, which makes it harder to identify the beginning and ending of each sentence and where to split them.

Moreover, longer sentences may lead to more factors associated to it, making it difficult to the classifier to learn the most important words for each class. We realized that the NLTK library has difficulty leading to this type of text. In this sense, we decided to test with other libraries in the second iteration.

Another issue was related to the labeling process. Some sentences were labeled by considering their implicit meaning obtained through the interpretation of the labeler. Consider, for instance, the following sentence: "*Firstly you should ALWAYS be able to tell who is signed in just by looking, so it should have the persons name showing sonewhere [sic] on the screen*". This sentence was assigned to the Usability label due to the interpretation of the labeler (i.e. the main researcher), as some usability guidelines cites the importance of always keeping the user informed where s/he is and what is happening in the application. However, the algorithm would not know about that. Instead, it should have been assigned to Improvement Request label, as the n-gram "should have" indicates something that needs to be improved, done, or added in the app.

6.3.5. Second Iteration

6.3.5.1. Tokenization and Manual Labeling

In this second iteration, we tested other three libraries to tokenize the reviews: TextBlob[10], Stanford CoreNLP[11], 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. A drawback of this library is that it can result in a greater number of sentences. For example, consider the following review "*I am amazed with this app but the only that is bad is that when u make your own account you can't change it anymore do could you please fix that but other than that everything else is awesome I recommend this app I hope you guys will download it.*". In this review, we have three factors associated to it: Attractiveness, Improvement Request, and Recommendation. When applying the NLTK, TexBlob, and CoreNLP tokenizers, they consider the review as a unique sentence. In this case, both labels are assigned to the entire review. By contrast, when applying Spacy, it results in four sentences: 1- "*I am amazed with this app but*" (Attractiveness); 2- "*the only that is bad is that when u make your own account you can't change it anymore do could you please fix that but other than that everything else is awesome*" (Improvement Request; Attractiveness);

[10] https://textblob.readthedocs.io/en/dev/
[11] https://stanfordnlp.github.io/CoreNLP/

3- “*I recommend this app*” (Recommendation); and 4- “I hope you guys will download it.” (None of the factors). By splitting into small sentences, it makes it easier for the classifier to learn the most important words for each factor. Due to this, we decided to use Spacy.

As the entire set had to be modified due to the different sentence segmentation approach, we had to restart the labeling process. In this second iteration, 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. Figure 6.4 presents the distribution of the training set by factor.

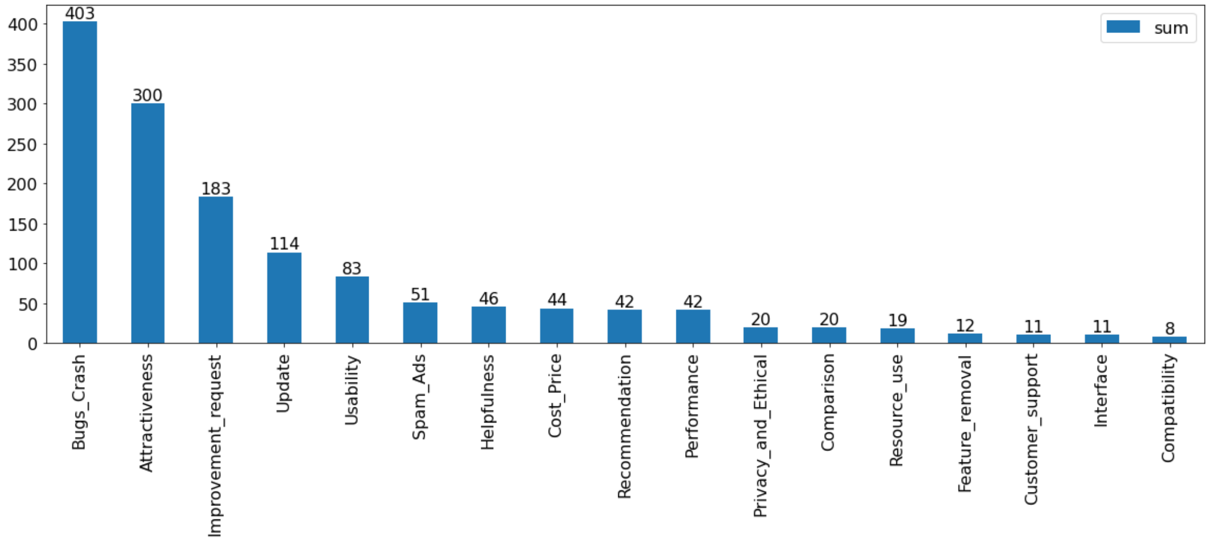


Figure 6.4 - Distribution of labeled instances by factor (second iteration).

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 5 random sentences from each class labeled by the main researcher (including sentences that were not assigned to any of the factors) and assessed the level of agreement by calculating the Cohen’s Kappa (Cohen, 1988) with another researcher, expert in HCI and UX. The results indicated a substantial agreement between the researchers (Cohen’s *d* = 0.663). We discussed the disagreements and identified improvement possibilities in the definition of some factors. 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 the 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 with respect to 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 with respect to some salient usability measure (e.g. learnability, performance, error rate, subjective satisfaction)*".

6.3.5.2. Preprocessing

In this second iteration, we reviewed the list of stopwords provided by 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. A previous work from Maalej and Nabil (2015) already indicated that the removal of the words from the corpora provided by NLTK might reduce the classification performance in particular categories such as bug reports. 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 training of the model.

Finally, due to the informal and noisy nature of the language used by end users, we also performed some additional preprocessing steps by using Natural Language Processing (NLP) tools as proposed by Palomba et al. (2017): spell correction and contraction expansion. The *spell correction* consists of replacing misspelled words according to the English vocabulary. To do so, we applied the symspellpy[12], a Python port of Symspell[13], an open-source spell correction algorithm. *Contraction expansion*, in turn, consists of replacing any contraction by its extended form (e.g. don't → do not).

6.3.5.3. Model Training and Testing

Regarding the training and testing parameters, we kept the same settings used in the first iteration: n-grams = (1,2), CountVectorizer and TF-IDF as feature extractors, and the four classifiers.

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

[13] https://github.com/wolfgarbe/SymSpell

6.3.5.4. *Results of the Second Iteration*

In this second iteration, we achieved promising results. Regarding the micro-averaged metrics, SVM obtained the best results due to an increase in recall, which resulted in a better F1-score than in the first iteration (see Table 6.3). Regarding the macro-averaged metrics, the increase in the number of instances let to greater scores in all three metrics. Finally, the extra step to perform TF-IDF still resulted in no improvement of the classifiers. As this extra step requires more time and does not improve the results, we decided to not apply it in the next iterations.

Table 6.3 – Results of each classifier with k-fold = 10 (second iteration).

	Micro			Macro			Fit Time
Classifier	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**	**(s)**
SVM (TF-IDF)	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 (TF-IDF)	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 (TF-IDF)	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 (TF-IDF)	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

6.3.6. Third Iteration

This iteration aimed to improve the macro-averaged metrics and identify the classifier that produces the best results to be used in UX-MAPPER. First, we increased the number of instances by focusing on factors with few samples to reduce the bias towards the largest factor: Accuracy, Comparison, Compatibility, Customer Support, Feature Removal, Interface, Privacy and Ethical, and Resource Use. To do so, we looked for keywords from already labeled sentences that would indicate its association to a given factor. For example, sentences from the "Resource Use" factor usually contain words such as "memory", "drain", "lot of", "space", and "bandwidth". We performed a manual search for these terms, analyzed the sentences and included them into the training set. In the end, we labeled 545 additional sentences, resulting in 1,677 labeled sentences, which distribution can be seen in Figure 6.5.

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 with another classifier called ComplementNB (CNB) proposed by Rennie et al. (2003) to tackle the low assumptions of MultinomialNB (MNB), being particularly suitable for imbalanced datasets, which is our case. We also tested with combinations of n-grams and with a classifier called XGBoost (eXtreme Gradient Boosting). It consists of a scalable and sparsity-aware machine learning algorithm that have been used in many machine learning and data mining challenges with good results (T. Chen and Guestrin, 2016). We aimed to analyze whether it provides better results in the context of app stores reviews mining. For this classifier, we set the learning parameter to "*softmax*", as it is designed to multiclass classification.

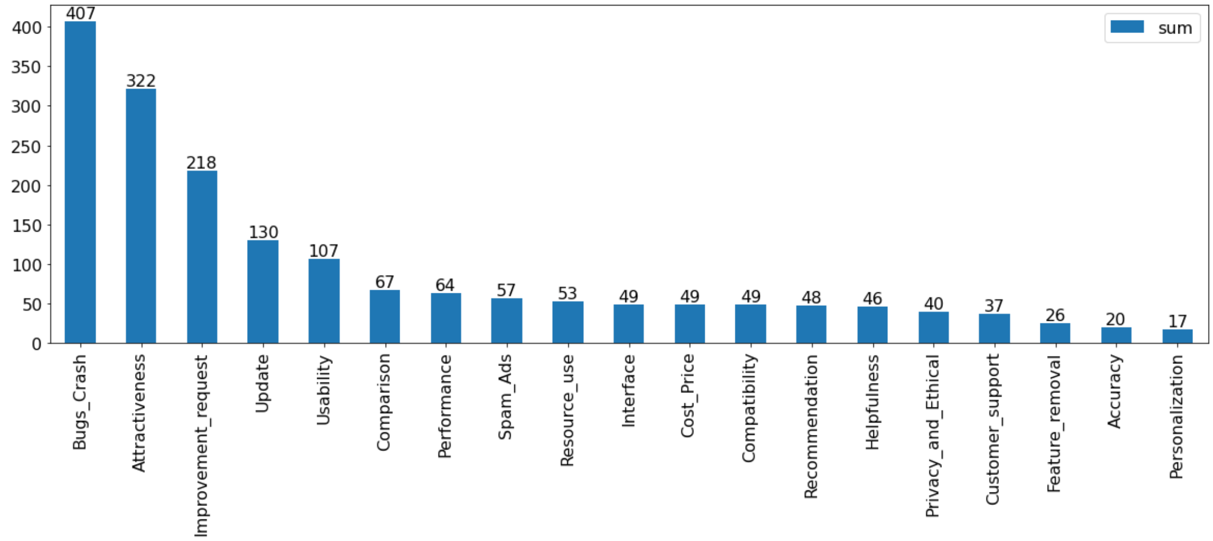


Figure 6.5 - Distribution of labeled instances by factor (third iteration).

6.3.6.1. Results of the Third Iteration

Overall, the results indicated that weighting the classes improved the performance of the classifiers, especially for Logistic Regression, which had a great improvement in recall (see Table 6.4). Regarding Naïve-Bayes, although ComplementNB performed better than MultinomialNB, it still obtained a poor overall performance. XGBoost, in turn, had comparable performance with SVM and LR. However, it required much more time to be executed.

In general, weighted SVM and weighted LR achieved the best results, with equivalent F1-scores in both micro and macro-averaged metrics. 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.

Table 6.4 - Results from the evaluation of the classifiers.

	Micro			Macro			Fit Time
Classifier	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**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	561.575

*class_weight = ‘balanced’

6.4. FEATURE EXTRACTION MODULE

In CHAPTER 5, we investigated the perception of participants regarding the proposal of a tool that extracts the most frequent terms from app reviews. The participants perceived the proposal of extracting the most frequent terms as useful for thinking of requirements. However, some terms extracted did not provide informative reviews, such as “waste time”, which only returned purely emotional comments that did not help identifying improvement opportunities. It was because the tool did not follow any type of criteria for selecting those terms. In order to provide more useful results, we searched for approaches from the literature that extracts features from app store reviews.

Dąbrowski et al. (2020) conducted a study with three state-of-the-art approaches that are widely known in Requirements Engineering community: SAFE (Simple Approach for Feature Extraction) (Johann et al., 2017), GuMa (Guzman and Maalej, 2014), and ReUS (Dragoni et al., 2019). De Araújo and Marcacini (2021) extended this study by proposing a novel approach using BERT, a pre-trained transformer network proposed by researchers at Google AI Language that is presenting state-of-the-art results for many NLP tasks, such as question answering, sentence classification, and sentence-pair regression (Reimers and Gurevych, 2019). Their approach, called RE-BERT, achieved the best results, followed by SAFE. Although we could choose RE-BERT due to its greater performance, we decided to investigate the outcomes of these approaches better by analyzing them through the eyes of practitioners from the industry. In this sense, we developed two versions of UX-MAPPER for being tested: one with SAFE and other with RE-BERT. We detail the characteristics and implementation of each approach in the next subsections.

6.4.1. Simple Approach for Feature Extraction (SAFE)

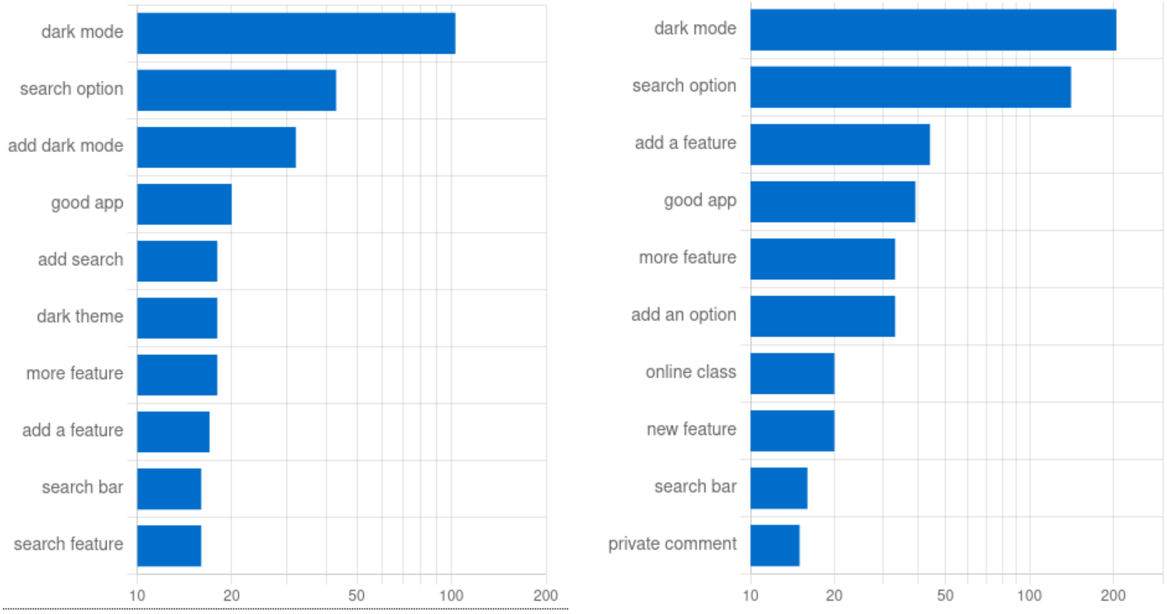
Proposed by Johann et al. (2017), it was designed to extract features from app descriptions and reviews from app stores. It extracts features based on Part-of-Speech (POS) patterns and sentence patterns. To identify these patterns, the authors used the Natural Language Tool Kit (NLTK) to assign POS-Tags to the description of 100 apps from Google Play store and performed a manual analysis. At the end of the process, they selected the patterns with at least 10 occurrences (see Table 6.5). Additionally, the approach identifies enumerations (comma-separated text) and conjunctions to identify lists of features (e.g., "send and receive attachments" are broken down into "send attachments" and "receive attachments"). It also performs a similarity matching to group similar features using cosine similarity.

Table 6.5 - POS-Tagging patterns from SAFE approach.

#	POS Pattern	Freq.	Example
1	Noun Noun	183	Group conversation
2	Verb Noun	122	Send message
3	Adjective Noun	119	Precise location
4	Noun Conjunction Noun	98	Phone or tablet
5	Adjective Noun Noun	70	Live traffic conditions
6	Noun Noun Noun	35	Email chat history
7	Verb Pronoun Noun	29	Share your thoughts
8	Verb Noun Noun	28	Enjoy group conversations
9	Verb Adjective Noun	26	Perform intuitive gestures
10	Adjective Adjective Noun	20	Super bright flashlight
11	Noun Preposition Noun	18	Highlight with colors
12	Verb Determiner Noun	14	Share an image
13	Verb Noun Preposition Noun	14	Use depth of field
14	Adjective Noun Noun Noun	12	Fast system virus scanner
15	Adjective Conjunction Adjective	12	Pre-installed and user-installed
16	Verb Preposition Adjective Noun	11	Choose from popular versions
17	Verb Pronoun Adjective Noun	11	Create your greatest album
18	Noun Conjunction Noun Noun	10	Song and artist album

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.

First, it splits the review into sentences by using SpaCy, a state-of-the-art open-source library for Natural Language Processing (NLP). Then, it preprocess the sentences by removing stopwords and applying lemmatization, which consists To group similar features, we used a clustering algorithm called “fast clustering”[14] 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. Figure 6.6 presents an example of the features presented by SAFE before and after applying Sentence-BERT. Terms such as “add dark mode” and “dark theme”, for instance, were grouped into “dark mode” feature, reducing the number of features to be analyzed.



(a) (b)

Figure 6.6 - Features from SAFE before (a) and after (b) applying Sentence-BERT.

6.4.2. RE-BERT

Proposed by de Araújo and Marcacini (2021), RE-BERT (Requirements Engineering BERT) extends the BERT model to extract software requirements. The BERT model can be fine-tuned to find significant correlations between the sequence of tokens in a review ($x = (x_1, x_2, \ldots, x_T)$) and a sequence of tokens that represents the software requirement ($x_a = (x_a1, x_2a, \ldots, x_Sa)$), where x_a is a subsequence of size S (with S >= 1) (de Araújo and Marcacini, 2021). Although BERT’s next-sentence prediction training strategy allows the model to learn general relationships between the review and the software requirement, known as global context, it usually fails to correctly identify software requirement tokens, as the global context is distant from software

[14] https://www.sbert.net/examples/applications/clustering/README.html#fast-clustering

requirement tokens. In this sense, the authors fine-tuned the model to focus on local contexts to increase the importance of tokens close to the software requirement (de Araújo and Marcacini, 2021).

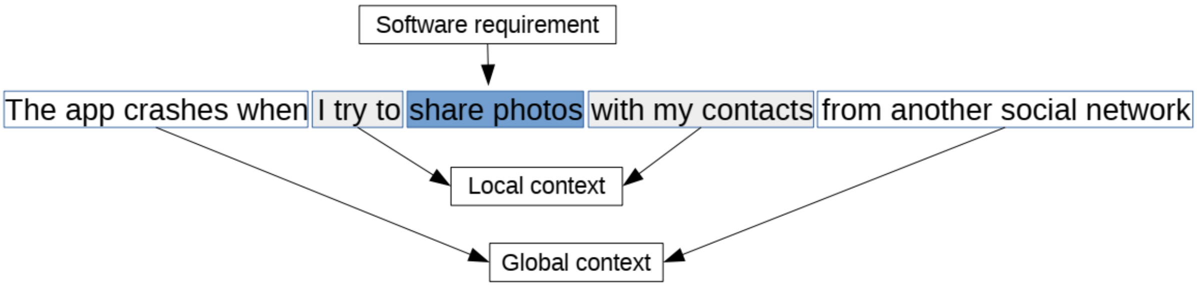


Figure 6.7 - Example of an app review divided into three parts: (1) software requirement; (2) global context; and (3) local context. Source: de Araújo and Marcacini (2021).

Figure 6.7 presents an app review divided into three parts. The global context describes the app's behavior and usage scenarios related to the "share photos" software requirement. The local context, in turn, is more associated to the actions, manners, and objects related to the software requirement, which the authors considered more important to identify software requirements properly.

The training set should be in the BIO (Beginning, Inside, Outside) format to train the model. The 'B' tag indicates that the token is the beginning of a software requirement. The 'I' tag indicates that the token is inside a software requirement. Finally, the 'O' tag indicates that the token is outside a software requirement. Considering the sentence below, we have the "raise hand option" as a software requirement. Thus, its tokens are assigned the 'B' and 'I' tags. All the other tokens are assigned the 'O' tag, as they are not part of any software requirement.

Please	add	the	**raise**	**hand**	**option**	in	the	android	version
O	O	O	**B**	**I**	**I**	O	O	O	O

Considering that our feature definition includes aspects related to UX, we trained the model with our own labeled dataset. To build the training set, we analyzed 3,000 reviews from three educational apps: Google Classroom, Programming Hub, and SoloLearn. First, we split the reviews into sentences using SpaCy, resulting in 6,113 sentences. Next, we analyzed each sentence by looking for features and tagging them using the BIO format. Finally, we trained RE-BERT with our training set.

6.4.3. Feature Extraction Performance Evaluation

After implementing SAFE and RE-BET, we evaluated their performance regarding three metrics: precision (P), recall (R), and F1. To do so, we followed the approach from Dąbrowski et al. (2020) and de Araújo and Marcacini (2021): let Γ be the set of words in a review sentence and $f_i \subseteq \Gamma$ be the set of words used to refer to feature i in that sentence. Two features $f_1, f_2 \subseteq \Gamma$ match at level n (with $n \in$ N) if and only if (i) the extracted feature is equal to or is a subset of the other, i.e. $f_1 \subseteq f_2$ or $f_2 \subseteq f_1$, and (ii) the absolute length difference between the features is at most n, i.e. $||f_1| - |f_2|| \leq n$. In summary, a feature can match the truth set in three levels: 1) *exact match:* when the feature is precisely the same present in the truth set; 2*) partial match 1 (n = 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 (n = 2):* when part of the feature matches the truth set, and there is at most two words that do not match. Extracted features in which the number of words that do not match the truth set is greater than two were considered as 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 6.6 presents the results for each approach according to the matching levels.

Table 6.6 – Comparison between SAFE and RE-BERT.

	Exact Match (n=0)			**Partial Match 1 (n=1)**			**Partial Match 2 (n=2)**		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
SAFE	0.532	**0.976**	0.689	0.548	**0.977**	0.702	0.560	**0.978**	0.713
RE-BERT	**0.707**	0.917	**0.798**	**0.726**	0.923	**0.813**	**0.735**	0.927	**0.819**

RE-BERT achieved the best results, mainly on precision. It is mainly 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.

6.5. UX-MAPPER WEB APPLICATION

In this subsection, we present the UX-MAPPER Web application.

6.5.1. Technologies Adopted

For the front-end development, we used Bootstrap[15], one of the most popular front-end open-source toolkit to develop interface components using HTML, CSS and JavaScript. To integrate

[15] https://getbootstrap.com/

our machine learning model in Python to the Web, we adopted Flask[16] as 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 [17]for version control and Heroku[18], a Platform as a Service (PaaS) that allow developers to build, run, and operate applications in the cloud.

6.5.2. UX-MAPPER Usage Overview

In this subsection, we present an overview of how practitioners can use UX-MAPPER. Figure 6.8 presents the initial page of the UX-MAPPER with SAFE feature extraction approach. In this page, practitioners can select the app they want to analyze. In this example, we selected Edmodo[19], a popular Learning Management System.

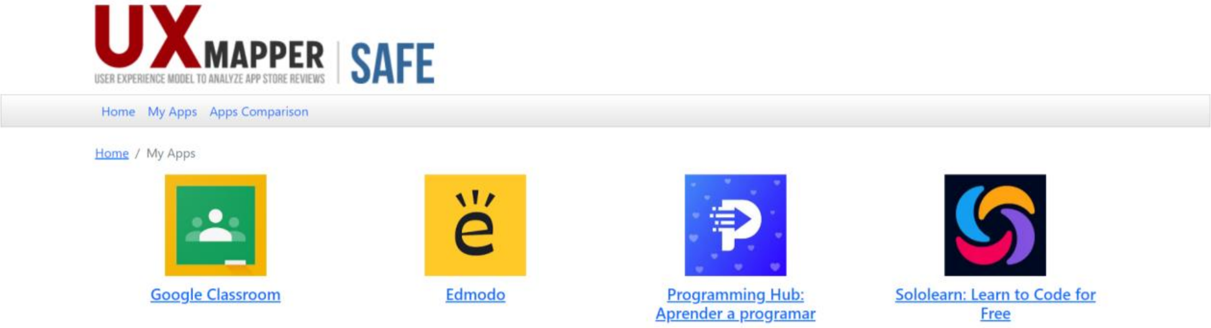

UX MAPPER
USER EXPERIENCE MODEL TO ANALYZE APP STORE REVIEWS
SAFE
Home
My Apps
Apps Comparison
Home / My Apps
Google Classroom
Edmodo
Programming Hub: Aprender a programar
Sololearn: Learn to Code for Free

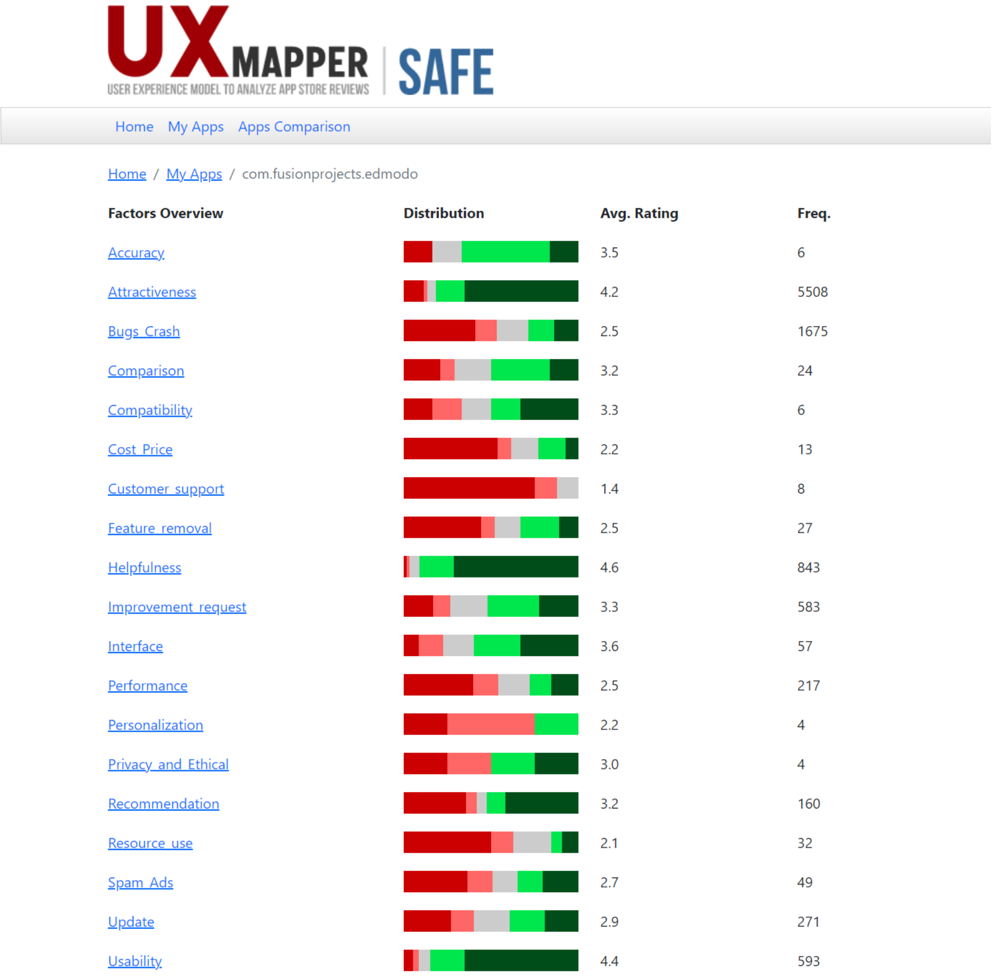
Figure 6.8 - Initial UX-MAPPER page.

After selecting the app, the practitioner is directed to a page that presents the set of factors, the distribution of star ratings (in which dark red represents 1-star rating, and dark green represents 5-star rating), the average rating of the factor, and the number of reviews associated with it (Figure 6.9).

Then, the practitioner can click on the desired factor to obtain the top 10 features extracted from the analysis of the reviews associated with this factor, as well as the distribution of the ratings according to the number of stars (Figure 6.10). The features are ordered by frequency, where the most frequent feature is presented on the top. From this screen,

[17] https://git-scm.com/
[18] https://www.heroku.com/
[19] https://new.edmodo.com/

practitioners can click on the bar related to the feature they want to have more details. By doing so, they are directed to a new page that presents the reviews associated with this feature (Figure 6.11). The reviews are ordered by relevance by default, i.e., reviews with the greatest number of thumbs up given by other users appears first on the top, similar to the Google Play Store.

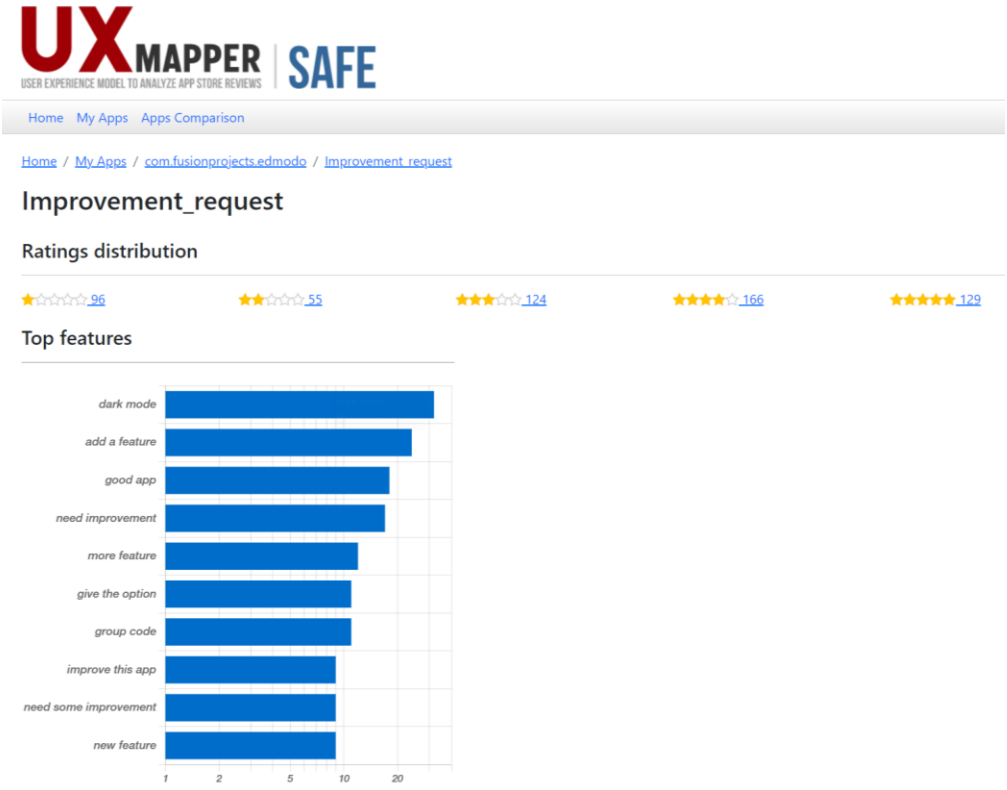


Home / My Apps / com.fusionprojects.edmodo

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
Cost_Price		2.2	13
Customer_support		1.4	8
Feature_removal		2.5	27
Helpfulness		4.6	843
Improvement_request		3.3	583
Interface		3.6	57
Performance		2.5	217
Personalization		2.2	4
Privacy_and_Ethical		3.0	4
Recommendation		3.2	160
Resource_use		2.1	32
Spam_Ads		2.7	49
Update		2.9	271
Usability		4.4	593

Figure 6.9 - Factors investigated by UX-MAPPER and their associated data.

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. Regarding "Improvement request" factor, a feature with a greater number of reviews with 1 or 2 stars, for example, 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) indicates 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, the practitioner 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 (Figure 6.12).



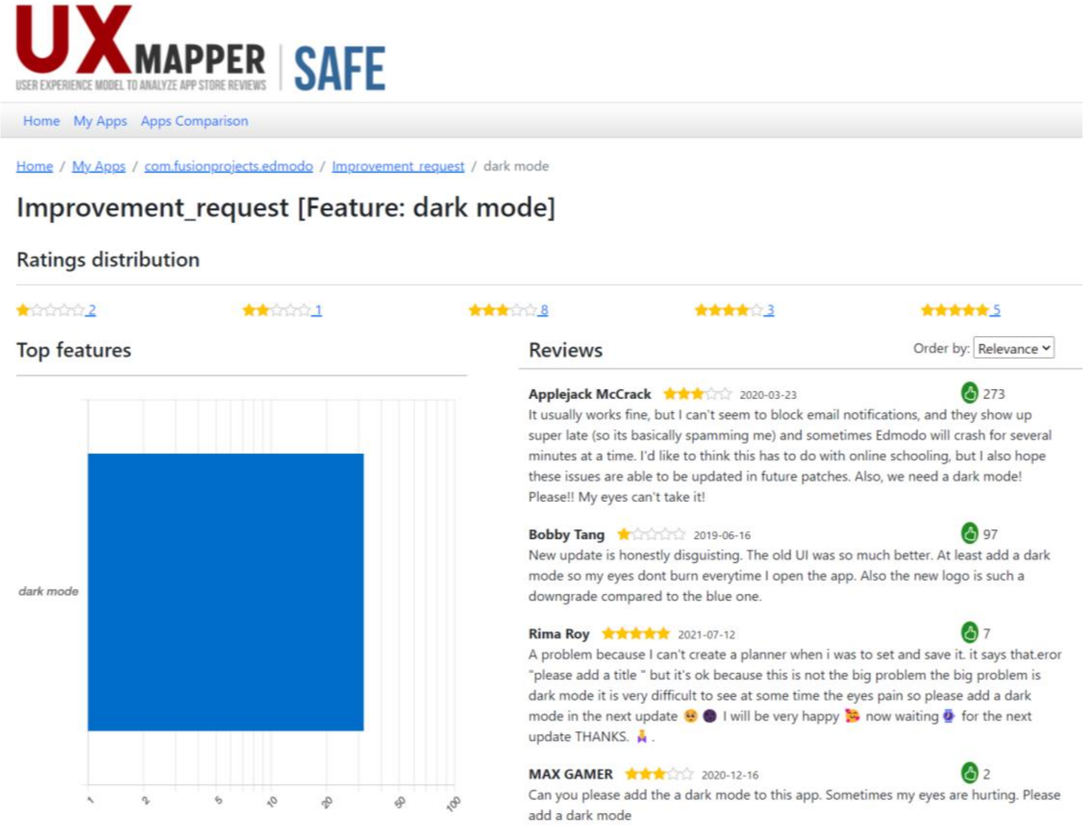
Improvement_request

Ratings distribution

★☆☆☆☆ 96 ★★☆☆☆ 55 ★★★☆☆ 124 ★★★★☆ 166 ★★★★★ 129

Top features

Figure 6.10 - Features extracted from the reviews associated with the Improvement Request factor.


UX MAPPER
USER EXPERIENCE MODEL TO ANALYZE APP STORE REVIEWS
SAFE
Home My Apps Apps Comparison
Home / My Apps / com.fusionprojects.edmodo / Improvement_request / dark mode
dark mode
1
2
5
10
20
50
100
Order by: Relevance

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.

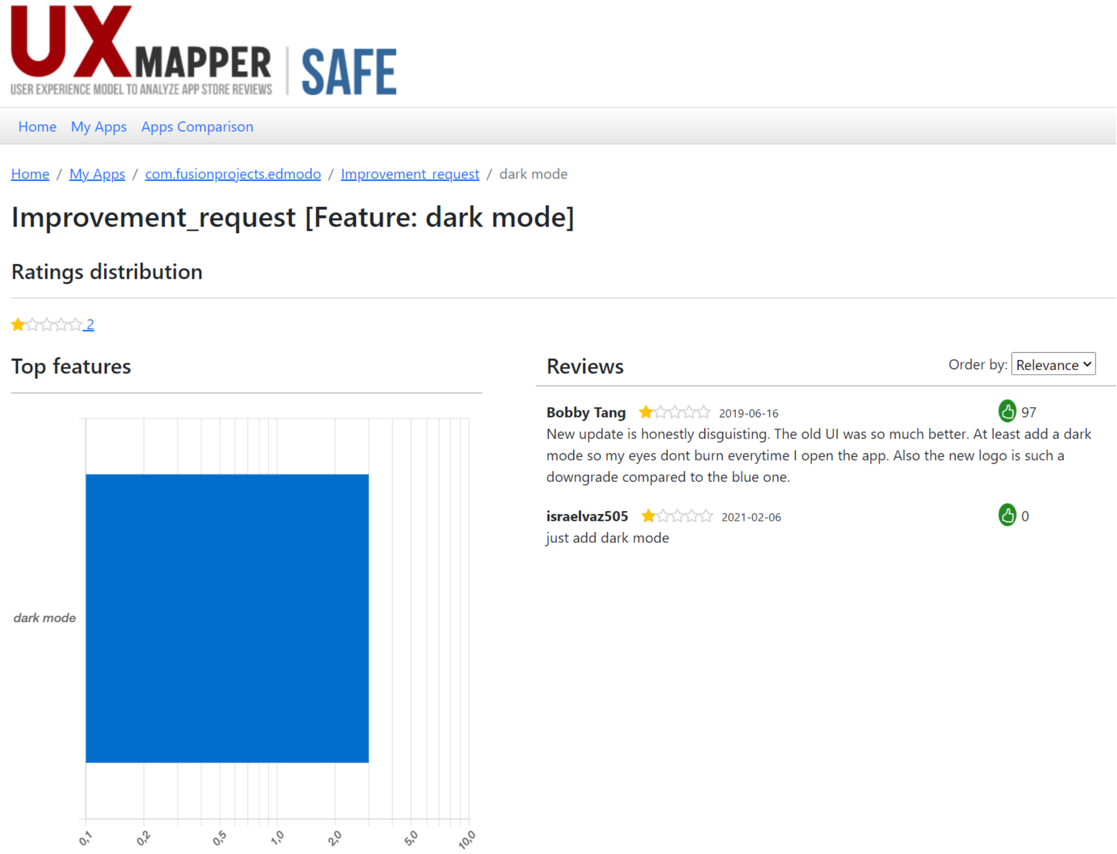
Rima Roy ★★★★★ 2021-07-12 7

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

Figure 6.11 - Reviews associated with the "dark mode" feature.



UX MAPPER | SAFE

USER EXPERIENCE MODEL TO ANALYZE APP STORE REVIEWS

Home My Apps Apps Comparison

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

Improvement_request [Feature: dark mode]

Ratings distribution

★☆☆☆☆ 2

Top features

Reviews

Order by: Relevance

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.

israelvaz505 ★☆☆☆☆ 2021-02-06 👍 0

just add dark mode

Figure 6.12 - Reviews filtered by the number of stars.

CHAPTER 7 – EVALUATING UX-MAPPER FROM PRACTITIONERS' PERSPECTIVE

This chapter presents the results of an empirical study conducted with practitioners from the industry to assess the usefulness and acceptance of UX-MAPPER and the relevance of the outcomes from the two selected feature extraction approaches.

7.1. INTRODUCTION

In the previous chapters, we presented the architecture of UX-MAPPER, the steps we followed to develop and refine it, and an overview of its functioning. In this chapter, we present the second iteration over the Design Cycle. We describe the steps we followed in planning the study with practitioners from the industry, the changes in the study's design, and the results we obtained. Due to the pandemic scenario of COVID-19 between March 2020 and February 2022, we conducted all the studies during this period remotely through Google Meet.

7.2. FIRST PILOT STUDY

In this pilot study, we aimed to put the planning to the test. We assessed the time required to conduct the study, whether the outcomes provide useful information to answer our research question, and whether there is a need for further adjustments. We present details of this study in the next subsections.

7.2.1. Participants and Materials

We conducted this pilot study with two participants selected by convenience with experience in requirements elicitation. Both had experience in requirements engineering, held the role of project manager, and had between one and three years of experience in this role. None of them had experience in analyzing user reviews. One participant worked with educational apps as part of their work, and the other participant had worked on at least one project in the educational apps' domain.

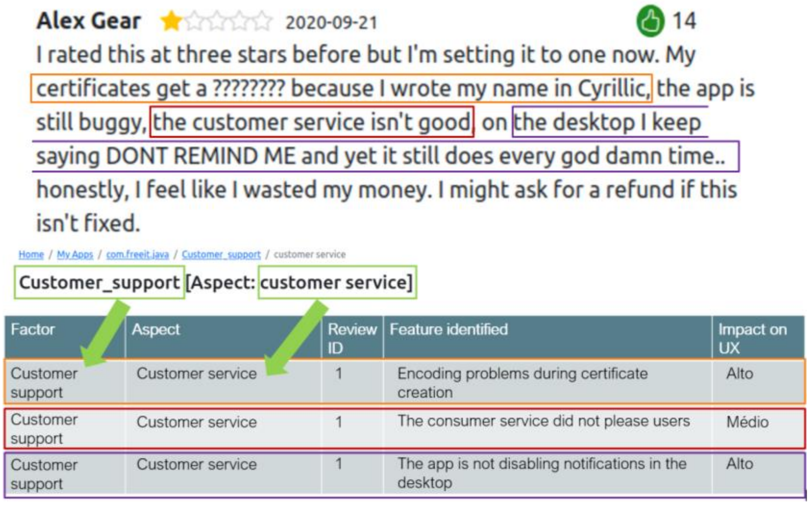
We used the following materials: i) an informed consent form (research project approved by the ethics committee of the Federal University of Amazonas - UFAM - Certificate of Presentation for Ethical Consideration–CAAE number 40928120.6.0000.5020); ii) a characterization questionnaire; iii) a spreadsheet to extract features; iv) the UX-MAPPER tool; and v) a post-study questionnaire comprising the core TAM constructs (Perceived Usefulness, Perceived Ease of Use, and Behavioral Intention) and additional TAM3 (Venkatesh and Bala, 2008) constructs (Job Relevance, Output Quality, Results Demonstrability), selected to evaluate the outcomes of UX-MAPPER and its potential to support the software development/improvement process (see **Table** 7.1).

Table 7.1 - Questions from the TAM3 questionnaire used in this study.

Dimension	Item	Description
Perceived Usefulness	PU1	Using UX-MAPPER improves my performance in software development
	PU2	Using UX-MAPPER increases my productivity in software development
	PU3	Using UX-MAPPER increases my effectiveness in software development
	PU4	I find UX-MAPPER useful in software development
Perceived Ease Of Use	PEOU1	My interaction with UX-MAPPER is clear and understandable
	PEOU2	Interacting with UX-MAPPER does not require a lot of my mental effort
	PEOU3	I find UX-MAPPER to be easy to use
	PEOU4	I find it easy to make UX-MAPPER do what I want it to do
Behavioral Intention	BI1	Assuming I had access to UX-MAPPER, I intend to use it
	BI2	Given that I had access to UX-MAPPER, I predict that I would use it
Job Relevance	JR1	In software development, using UX-MAPPER is important
	JR2	In software development, using UX-MAPPER is relevant
	JR3	The use of UX-MAPPER is pertinent to various activities related to software development
Output Quality	OQ1	The quality of the results I get from UX-MAPPER is high
	OQ2	I have no problem with the quality of the UX-MAPPER results
	OQ3	I rate the UX-MAPPER results as excellent
Results Demonstrability	RD1	I have no difficulty telling others about the results of using UX-MAPPER
	RD2	I believe I could communicate to others the consequences of using UX-MAPPER
	RD3	The results of using UX-MAPPER are apparent to me.
	RD4	I would have no difficulty explaining why using UX-MAPPER may or may not be beneficial

7.2.2. Procedure

We began asking the participants to read and sign the informed consent form and fill the characterization questionnaire. Next, we contextualized the research and the motivation for the study. After that, we presented a brief tutorial on how to use UX-MAPPER and provided a scenario to contextualize the tasks the participants should perform in the study. In this scenario, the participant is a software engineer from a company that has Google Classroom as their flagship. The participant's goal is to analyze its reviews and extract features that would support the development of a new release focusing on bringing users a positive UX. Next, we provided the definition of feature and an example of how to extract it from a user review and fill in the extraction spreadsheet (see Figure 7.1). We defined feature as an attribute of an app that is provided intentionally that can affect the UX. A feature can: i) be used to perform a task ("view contact") by a user; ii) be seen as an application functionality (e.g., "send message"), a module (e.g., "user account") providing functionality (e.g., "delete account" or "edit information") or a design component (e.g., UI) providing functional features (e.g., "configuration screen", "button"); iii) be seen as an expression (e.g., "the calendar sometimes gets out of sync with the server"). Finally, we provided a set of tasks that the participants should perform (see Table 7.2). The participant should analyze the reviews associated with the aspect extracted from the Attractiveness, Bugs/Crash, and Improvement request factors. We selected these factors as they are the ones with the greatest number of reviews associated. Due to time constraints, we decided that the participant should explore the first three aspects and the first five reviews from each factor, which results in a total of 45 reviews to be analyzed. Each participant should perform the same set of tasks for each of the approaches (SAFE and RE-BERT) in a cross-over design. To avoid the primacy bias, i.e., the over-weighted influence of the first experience (Shteingart et al., 2013), and the training effect, each participant began with a different approach. Each participant participated in the study individually and on different days.



Alex Gear ★☆☆☆☆ 2020-09-21 14

I rated this at three stars before but I'm setting it to one now. My certificates get a ???????? because I wrote my name in Cyrillic, the app is still buggy, the customer service isn't good on the desktop I keep saying DONT REMIND ME and yet it still does every god damn time.. honestly, I feel like I wasted my money. I might ask for a refund if this isn't fixed.

Home / My Apps / com.freeit.java / Customer_support / customer service

Customer_support [Aspect: customer service]

Factor	Aspect	Review ID	Feature identified	Impact on UX
Customer support	Customer service	1	Encoding problems during certificate creation	Alto
Customer support	Customer service	1	The consumer service did not please users	Médio
Customer support	Customer service	1	The app is not disabling notifications in the desktop	Alto

Figure 7.1 - Example of how to identify and extract features.

Table 7.2 - Script followed by the participants.

ID	Task description
1	Access the home page of UX-MAPPER.
2	Select the Google Classroom app.
3	Explore the following factors: Attractiveness, Bugs/Crash, and Improvement request.
4	Filter the reviews with one star and order them by relevance (number of thumbs up the review received).
5	Explore the first three aspects and identify features that could affect UX in the first five reviews.
6	Classify each feature according to your perception of its impact on UX (none, low, medium, and high).

7.2.3. Results

The results revealed issues in our initial planning already with the first participant. The first critical issue was the required time. The first participant took 1 hour and 45 minutes to extract features from the 45 reviews returned by the RE-BERT approach (5 reviews from 3 aspects from 3 factors). Such required time makes the study unfeasible to be conducted with practitioners from the industry, considering they have limited availability. Thus, we decided not to perform a cross-over design and 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 a reduced number of reviews, the participant took 1 hour and 8 minutes to extract features from this approach, which was still high to perform a cross-study design. When analyzing the features extracted, we also realized that they are not directly comparable, as the aspects extracted by the approaches lead to different reviews that resulted in different features. Moreover, we were not making direct questions about the approach itself. Thus, we could not identify which of them were better from the participant’s point of view.

7.3. SECOND PILOT STUDY

We conducted a second pilot study to put our planning to the test again. Our goal was to assess the adequacy of our planning after making the adjustments based on the results from the first pilot study. Each participant visualized the outcomes of both approaches, but only interacted with one of them to reduce the time required to conduct the study (details in the next subsection). Additionally, instead of identifying features from the reviews (the most time-consuming task), each participant assessed whether the aspects extracted by the approach are understandable and useful for identifying potential features to improve the UX. To do so, we changed the tool so that the reviews were presented only after clicking on the aspect. We made this change for the participant to reflect on the meaning and usefulness of the aspect itself before visualizing the reviews.

7.3.1. Participants and Materials

We conducted this pilot study with four participants with experience in requirements elicitation, selected by convenience. We had the following profiles in this study: two Developers, a Tester, and a Requirements Engineer/P.O. Three participants had between one and three years of experience in their roles, and one participant had between four and six years of experience. None of them had experience in analyzing user reviews, but all had already worked with or used at least one educational app.

We used the following materials in this study: i) an informed consent form; ii) a characterization questionnaire; iii) a presentation for the participant to assess the aspects extracted by each approach (details in the next subsection); iv) the UX-MAPPER tool; and v) the post-study questionnaire applied in the first pilot study.

7.3.2. Procedure

First, we introduced the context of our study and its motivation. Then, we presented a brief tutorial of how to use the tool and the usage scenario. Next, we instructed the participant to interact with one of the approaches (RE-BERT or SAFE) at random. We asked the participant to analyze each aspect and reflect on whether it would be useful to improve the UX of the app. Then, the participant was allowed to click on the aspect and visualize the reviews associated with it. After visualizing the reviews, we presented the aspects extracted by both the approach the participant interacted with and the other, side by side in a PowerPoint presentation, for each of the three factors (Attractiveness, Bugs/Crash, and Improvement request). We asked the participant to analyze each aspect and reflect on their meaning to assess whether it is understandable and the potential to return helpful information to improve the UX of the Google Classroom app. We crossed out the aspects that the participant did not consider understandable or useful during the study (see Figure 7.2). After assessing all aspects from all factors, we asked the participant to decide which approach s/he would choose to use. Finally, we asked the participant to answer the post-study questionnaire.

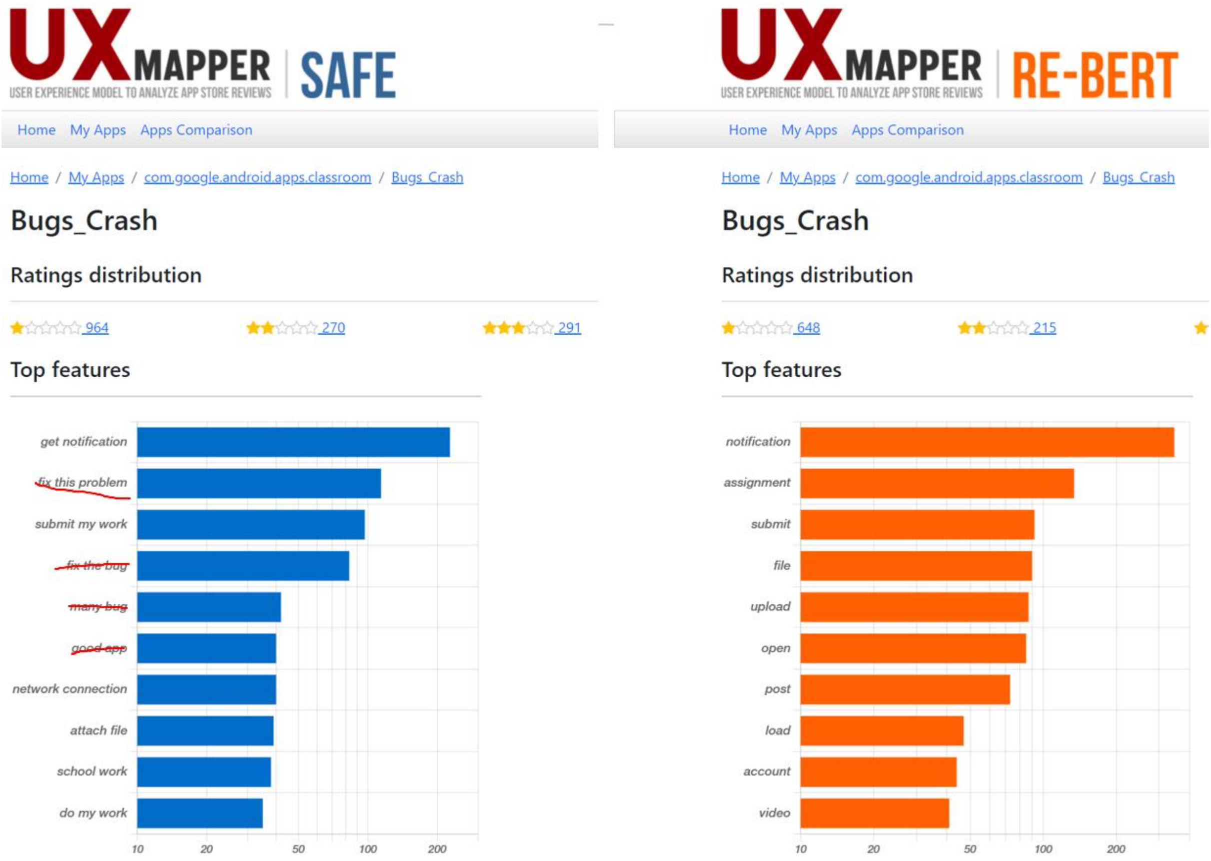


Figure 7.2 - Aspects considered not understandable or useful by participant P2.

7.3.3. Results

The results of this second pilot study allowed us to have a more direct comparison between the approaches. By crossing out the aspects considered not relevant or not understandable, and asking the participants the motives for their removal, it was possible to compare the approaches and understand the reasons behind their preferences.

Participants P1 and P3 preferred RE-BERT because they considered that it provided more specific aspects, i.e., aspects that present functionalities that could be added, improved, or fixed (e.g., upload, post, video). Participant P2 preferred SAFE because it provides more context by including more terms, such as “dark mode” and “search option” instead of just “mode” and “option” like RE-BERT. Finally, participant P4 considered both very similar, with a slightly preference towards RE-BERT because it provided some more straightforward aspects.

To understand the results better, we divided the participants into two groups according to the approach they interacted with. Figure 7.3 presents the number of relevant aspects in each factor by approach. In general, the RE-BERT approach had more aspects considered relevant than SAFE. The results indicate a possible correlation between the number of aspects considered relevant 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 make some adjustments for the final study to avoid this bias. We also felt the need for more straight-to-the-point metrics, rather than just the number of relevant aspects, to make a more thorough comparison between the approaches.

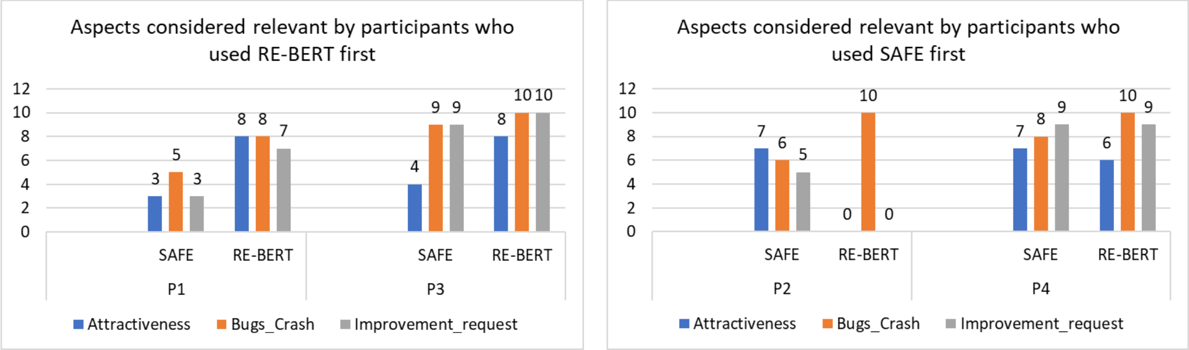


Figure 7.3 - Number of aspects considered relevant by the participants.

7.4. FINAL STUDY

The two pilot studies were essential to identify the feasibility of the planning and potential issues that could affect the results. To reduce the primacy bias, i.e., the over-weighted influence of the first experience (Shteingart et al., 2013), we decided to present the aspects of both approaches side by side before the participant interacts with the tool. After analyzing the aspects, the participant interacted with both approaches to explore the aspects they did not understand or considered irrelevant. We also added three Likert-type questions to directly evaluate the usefulness, easiness, and diversity of the aspects extracted by each approach (see Table 7.3), as well as an open question to justify their answers and get qualitative data.

Table 7.3 – Questions included in the post-study questionnaire.

ID	Question description
1	The features presented by the [SAFE/RE-BERT] approach are informative and make it possible to identify opportunities for UX improvement".
2	The features extracted by the [SAFE/RE-BERT] approach are varied and unique.
3	I have no difficulty understanding the features extracted by the [SAFE/RE-BERT] approach.

7.4.1. Participants and Materials

We conducted the study with 14 practitioners who did not participate in the pilot studies, selected by convenience. To reflect the target audience of our tool, we recruited only participants that work in the software development industry with experience in requirements engineering. Experience in analyzing user feedback was desirable but not mandatory. Most of them held the role of Requirements Engineer/Product Owner and Developer (Figure 7.4a) and all the participants had at least one year of experience in their roles (Figure 7.4b). Regarding their experience in analyzing user reviews/feedback, the majority of the participants had performed this type of analysis in at least of project (Figure 7.5a). Only one participant did not have experience in the educational apps' domain (Figure 7.5b). Finally, three participants had already used automated text analysis approaches (**Figure** 7.6).

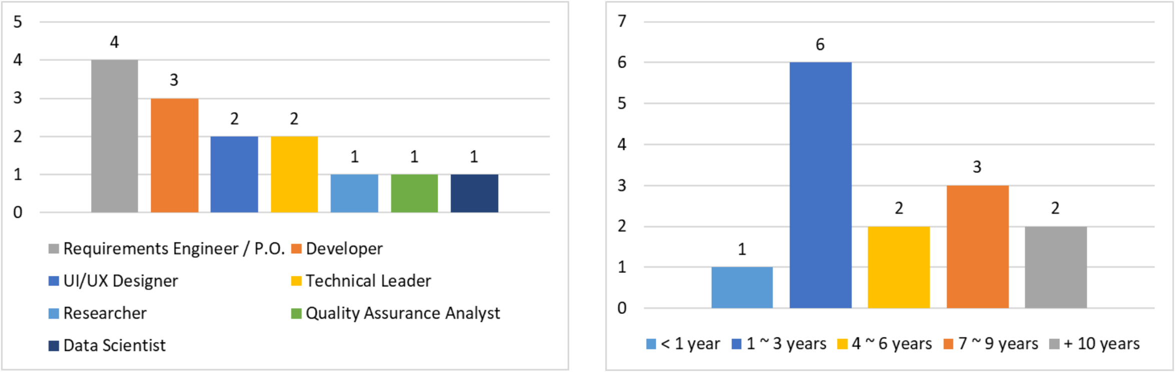
In this study, we used the following materials: i) an informed consent form; ii) a characterization questionnaire; iii) a presentation for the participant to assess the aspects extracted by each approach; iv) the UX-MAPPER tool[20, 21]; and v) a post-study questionnaire to assess the approaches, comprising the Likert-type questions presented in Table 7.2; 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.

7.4.2. Procedure

The procedure was similar to the previous studies. First, we introduced the context of our research and the motivation for conducting the study. Then, we asked the participant to sign the informed consent form and fill the characterization questionnaire. Next, we presented the aspects extracted by both SAFE and RE-BERT approaches for each of the three factors (Attractiveness, Bugs/Crash, and Improvement request) side by side. We asked the participant to analyze each aspect and reflect on their meaning to assess whether it is understandable and the potential to return helpful information to improve the UX of the Google Classroom app. Aspects that were not considered understandable or useful were crossed out similar to the second pilot study. After assessing the aspects, the participant had to decide which approach they would choose to use. Next, we asked the participants to explore the reviews of each aspect from both approaches, focusing on the aspects they crossed out previously. The participants could explore the tool freely to reflect on their actual usage in a real situation. This exploration aimed to investigate whether their opinion on their preferred approach changes by reading the reviews and understanding the aspects better. After exploring both approaches implemented in the UX-MAPPER tool, we asked the participant whether s/he would change his/her opinion on

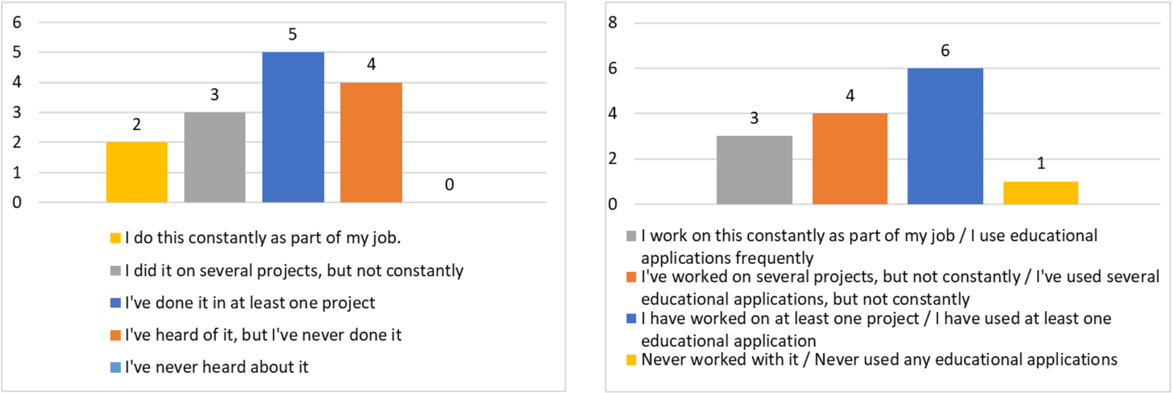
[20] https://ux-mapper-safe.herokuapp.com/myapps
[21] https://ux-mapper-rebert.herokuapp.com/myapps

their preferred approach. Finally, we asked the participants to answer the post-study questionnaire.



(a) (b)

Figure 7.4 – Participants profile by: (a) role and (b) years of experience in this role.



(a) (b)

Figure 7.5 – Participants experience: (a) in analyzing user feedback and (b) on educational apps domain.

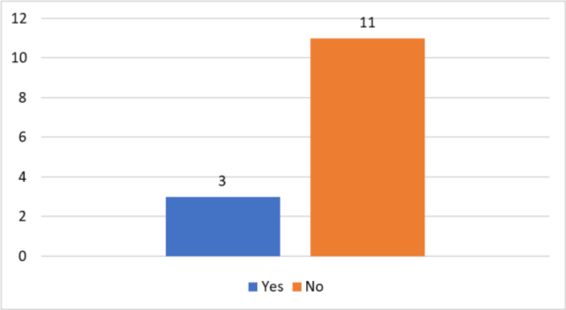


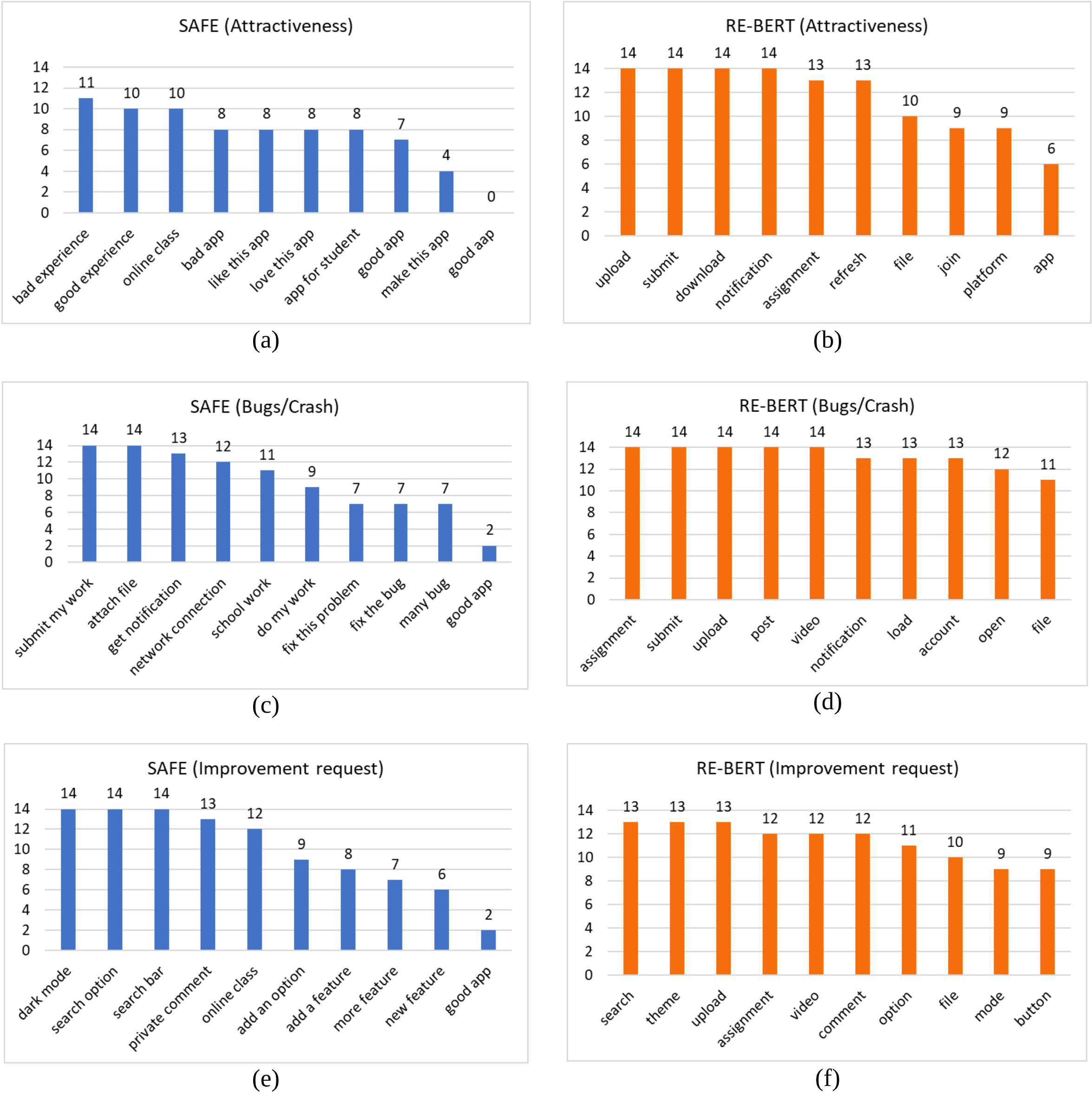
Figure 7.6 – Number of participants who already used and did not use automated text analysis approaches.

7.5. RESULTS

We divided the results into two subsections to facilitate understanding. First, we present the results for the comparison of the approaches. Next, we present the results for UX-MAPPER.

7.5.1. SAFE vs RE-BERT

The first criteria we adopted to compare the approaches was the relevance of the terms, which we assessed by asking the participants to cross out the terms they considered irrelevant. Figure 7.7 presents the number of participants that considered a given aspect relevant by approach and factor. In general, RE-BERT had more aspects considered relevant than SAFE.



(a) (b) (c) (d) (e) (f)

Figure 7.7 – Number of participants that considered the aspect relevant by approach and factor.

“Attractiveness” was the factor with less relevant aspects in SAFE. All the participants identified the “good aap” typo and considered merging it with “good app” aspect. Thus, we considered it as “not relevant”. Most of the participants did not understand the aspect “make this app” and considered it irrelevant. Some participants pointed out that aspects such as “good app”, “bad app”, “love this app” do not provide useful information, as they are only general opinions about the app. Participant P13, for instance, stated "*these aspects will not help me think of something related to Attractiveness, they are just indicating users' opinion about the app as a whole*". Conversely, most of the participants considered "bad experience" and "good experience" as relevant, although they are also general terms like the ones we mentioned before. Participant P4 stated "*regarding experience, I would check it, as it may have some features that 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*". It highlights that the participants find it essential to analyze what experience the app is conveying to users. Regarding RE-BERT, we had similar results for general terms. The "app" aspect, for instance, was considered irrelevant by most of the participants.

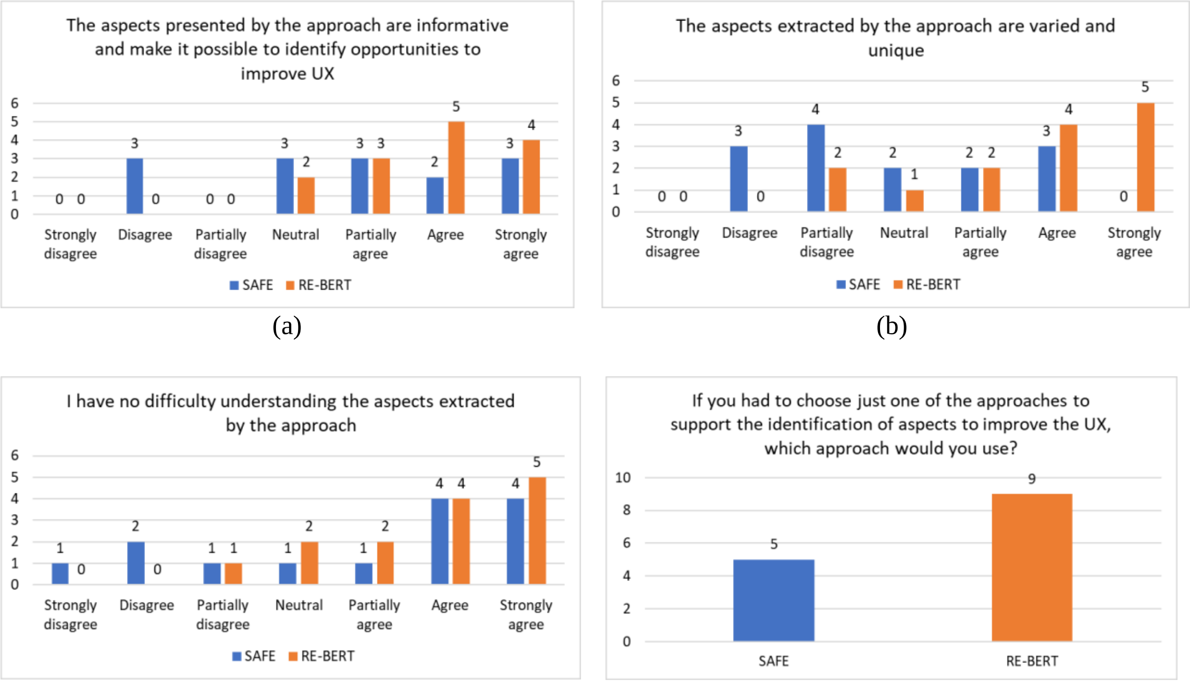
RE-BERT had the best performance in "Bugs/Crash" factor, as the aspects extracted are more specific and mainly related to functionalities. In turn, SAFE had the poorest performance in this factor. Many participants considered the aspects generic and redundant. Terms such as "fix the problem", "fix the bug", and "many bug" are very similar and do not point out to specific issues that need to be checked. Participant P2, for instance, stated "*if I say 'fix this problem' regarding a bug or a crash, what is exactly the problem? It is too generic and not useful unless you go see the reviews*". The participants also considered it awkward to have "good app" as an aspect from Bugs/Crash factor. After visualizing the reviews, they realized that it was because many users praised the app at the beginning of the review, 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 related to "Bugs/Crash", that is, the issues. Participant P3, for instance, stated "*the aspect 'good app', for example, refer to an unimportant part of the problem. In this case, what is important is what comes next, like 'good app, but…'*".

Finally, SAFE achieved its best result in "Improvement request" factor, although still below RE-BERT's performance. The participants considered that it provided more context by using more terms instead of only one as the RE-BERT approach. For example, SAFE returned the "dark mode" aspect, while RE-BERT only returned "mode", which makes it difficult to understand what the latter is referring to. Participant P10, for instance, considered all the aspects returned by RE-BERT as 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 throwed these words and I don't know what to do with them*". However, SAFE still presented generic terms, such as "add a feature", "more feature", "new feature", and "good app”, which many participants considered irrelevant. Participant P8 said “*what feature the user is asking me to add? These aspects are not bringing me improvement information*”.

Regarding the first question from the post-study questionnaire about the amount of information provided by the aspects extracted, the results indicated that, in general, the participants found the aspects presented by RE-BERT more informative to identify opportunities to improve UX (Figure 7.8a). Participants reported that RE-BERT found more aspects and that they are more specific, concise, and assertive than those extracted by SAFE. Participant P3, for instance, stated “*RE-BERT seems to be more assertive regarding the specification of the features based on user reviews. I think SAFE ended up considering the weight of the most recurring words so much that it did not consider the view of a feature*”. However, some participants felt the need for more aspects related to sentiments, while others reported that some aspects are generic. Regarding SAFE, some participants considered that the aspects it extracted are more related to UX, in addition to be 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*”. Conversely, some participants considered this subjectivity a drawback, as they only express users' opinion. Participant P14, for instance, stated “*probably reviews with 'good app' would not have relevant information to be analyzed, as they can literally be only the word 'good app' [in the review]*”. Other participants also pointed out that there are aspects that are not clear enough or just some common irrelevant expressions, such as “fix this problem” and “good app”. Participant P6 reported “*SAFE returns some random snippets, which for an analysis becomes expendable*”. In turn, there was also participants who considered that these generic terms can be beneficial to identify other issues than those presented in the graph, which may be useful for exploratory purposes.

The second question aimed to assess whether the approaches provide varied and unique aspects (Figure 7.8b). The results indicate that the aspects extracted by SAFE are not diverse and unique compared to those from RE-BERT. The participants reported that some aspects from SAFE are redundant (e.g., “fix the problem” and “fix de bug”) or variations of the same aspect that do not provide much context (e.g., “good app” and “bad app”), which require the practitioner to analyze the comments to interpret them. Participant P11, for instance, stated “*I think SAFE brought more repeated features, which could be combined to represent something more significant*”. They also pointed out that its performance on “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.



(a) (b)

(c) (d)

Figure 7.8 - Responses to the questions comparing the approaches in the post-study questionnaire.

The third question aimed to assess the participants understanding of the aspects extracted (Figure 7.8c). In general, the majority of the participants did not have difficulty in comprehending the aspects 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 expression than aspects, and some of them are redundant. There were also participants pointing out that they had difficulty in understanding the aspect, but after reading the reviews, it made sense. Participant P2, for instance, stated "*initially the aspects from SAFE did not make much sense, but after visualizing the reviews, it was easy to identify why they were selected*". Conversely, it was also considered a drawback by some participants, as they wasted time visualizing the reviews to comprehend the aspect extracted. Participant P4, for instance, said "*there are some redundancies in SAFE, which makes it necessary to analyze the comments to understand the main points*". Regarding RE-BERT, the conciseness of the aspects and their focus on more objective aspects made the outcomes easier and clearer to understand. Participant

P8, for instance, stated "*the top aspects from RE-BERT already provide the idea of what is the problem of the app*". In turn, some participants reported that some aspects are generic, and that sometimes it is not so clear because it presents only one word, making it difficult for practitioners without knowledge of the domain to understand its meaning. Participant P3, for instance, stated "*although RE-BERT is more assertive (...), it is less clear because it only brings one word for consideration in the factor*". Participant P12, in turn, said "*I can understand [the aspects] due to my experience and knowledge in the area, but people who do not have the minimum [experience and expertise] would have difficulty*".

Finally, the fourth question asked which approach the participant would choose to use (Figure 7.8d). The majority of the participants preferred RE-BERT, which can be explained by its focus on functionalities and the more straightforward and less redundant aspects it provides. Participants who considered more important to get more subjective and emotional aspects, in turn, preferred SAFE.

7.5.2. UX-MAPPER

In general, the results from the TAM questionnaire indicated a positive acceptance of UX-MAPPER (see Figures Figure 7.9 and Figure 7.10). Participants who had experience in using automated text analysis approaches tended to give slightly lower ratings for UX-MAPPER. Regarding Perceived Usefulness (PU), all participants considered that it is useful in the context of software development (PU4) by improving their performance (PU1) and increasing their effectiveness (PU3). 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 is the reflect of 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. Some participants had difficulty interacting with the tool, which can explain the lower ratings they provided. In the graph with the aspects extracted, for instance, 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 name of the category 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]*”.

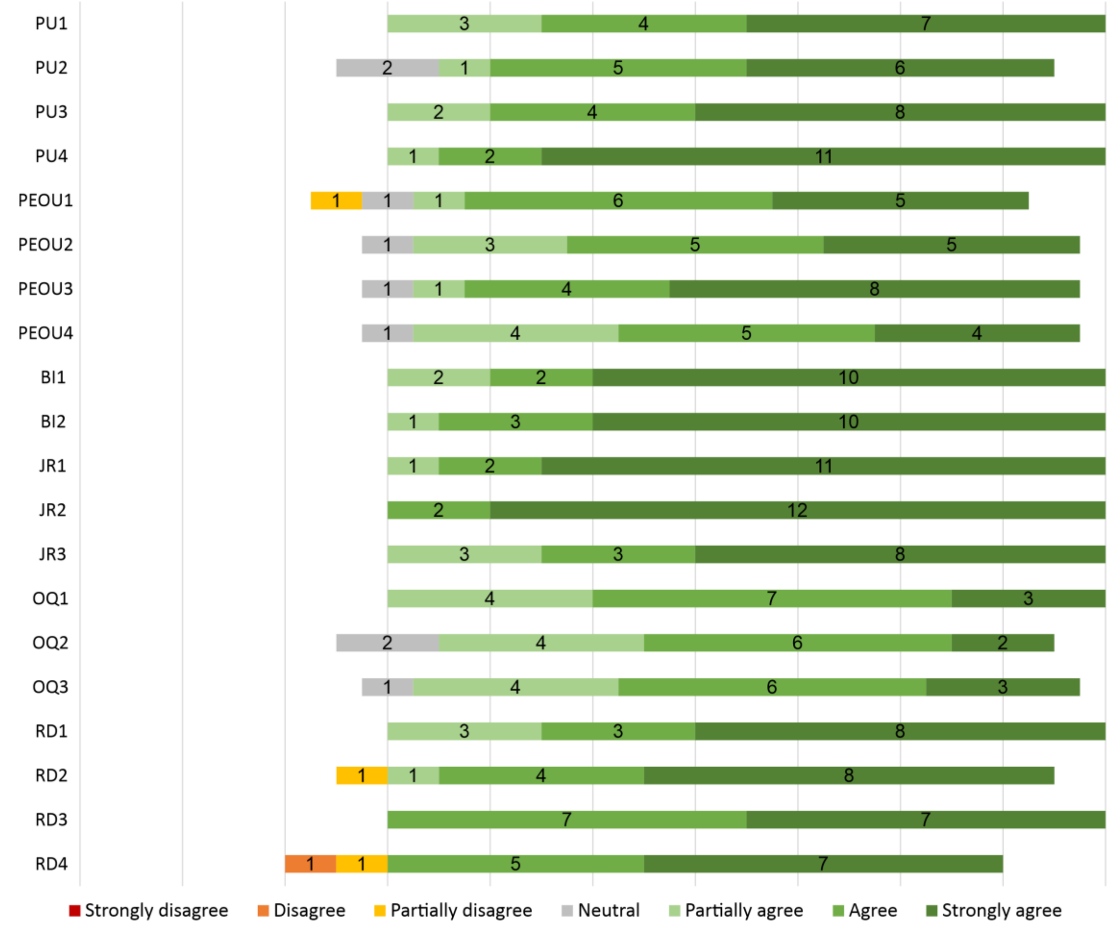


Figure 7.9 - Distribution of the responses for TAM3 questionnaire items.

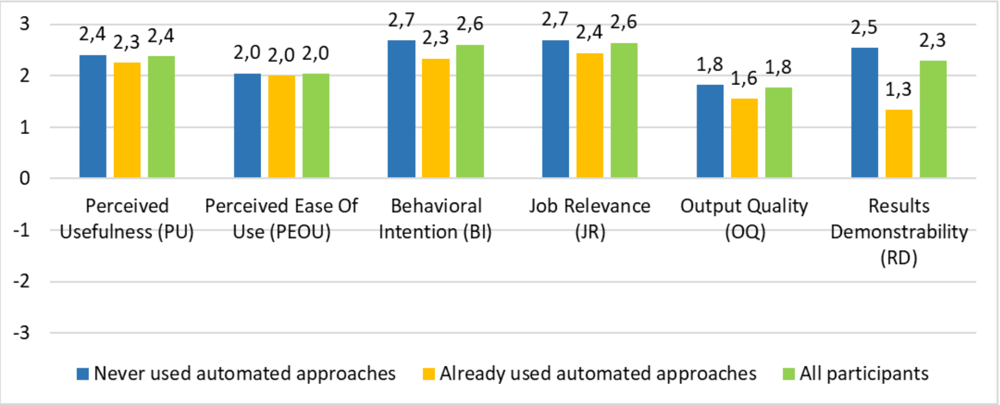


Figure 7.10 - Mean of each dimension of TAM3 questionnaire according to the experience in using automated approaches.

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 result reflects their perceptions of its usefulness and ease of use, which, according to Davis et al. (1989), predicts actual system usage.

When asked about its relevance on their jobs (JR), all participants were unanimous in affirming that UX-MAPPER is relevant. For instance, participant P9, who is a UX Designer that worked in 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. Although they considered the tool useful and relevant for their job, this result highlights that there is 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.

Finally, Results Demonstrability (RD) was the lowest rated dimension among the participants with experience using automated text analysis approaches. Such results indicate that the benefits of using UX-MAPPER is 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 there were much redundance and generic terms in SAFE. She was also the only participant who interacted from a smartphone, which behavior was not tested prior to 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 on the meaning of the terms from RE-BERT. For instance, in Attractiveness factor, she had to speculate whether "upload, submit, download" are being talked positively by users or not, or what users are talking 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.

7.6. DISCUSSION

The results of this study revealed varied preferences of practitioners when analyzing user reviews. In general, they desire to have an overview of the main points they need to look at to improve the app and meet users' needs. Thus, they prefer terms that are more specific to features and functionalities instead of generic terms that require further analysis of the reviews, which explains the greater preference for the RE-BERT approach. On the other hand, there were also participants that considered the generic terms useful in an exploratory setting, given that they group various reviews that may contain features that were not among the top features presented. Considering these two perspectives, UX-MAPPER could be used in two scenarios: 1) for improving the quality of existing apps, which features and functionalities are already known by the practitioners and they just want to focus on the most frequently mentioned ones; 2) for exploring the reviews of competing apps, for instance, to identify a set of features and functionalities that users are requesting, liking, or hating.

Regarding the features extracted, we identified that the level of the details required by practitioners to understand the aspect depends on the factor. For "Bugs/Crash" factor, for instance, practitioners did not mind having just one term extracted. This indicates that they prefer more straight to the point terms related to functionalities and that are not subjective. It is because identifying which functionalities are causing the bug is essential to make fixes, and generic or subjective terms do not help towards it. By contrast, some participants preferred to have details of what features or functionalities users are requesting in "Improvement request" factor. Generic terms such as "option" and "button", were not sufficient for them to identify the requested changes.

The results also revealed two stakeholder profiles. One is more concerned to subjective aspects related to what users are feeling, 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. In this sense, addressing both types of features is essential to provide a more holistic view of the experience and support practitioners identifying improvement opportunities from different perspectives.

Finally, the results from the TAM3 questionnaire revealed a positive acceptance of UX-MAPPER. The unanimity regarding 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. The classification of the reviews into factors and the features extracted help organize and find information quickly, which might increase productivity by reducing the effort to extract such information manually from the reviews.

In turn, participants who already used automated text analysis approaches considered that the benefits of using UX-MAPPER is not so apparent. Although they considered it useful and relevant for their jobs, the results indicate that there is still room for improvements, mainly in the outcomes of 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. The tool can also be improved. The lack of clues on whether the graph is clickable or not, the impossibility of clicking on the feature’s name, the unintuitive rating distribution graph in the factors overview, and other navigation problems indicate that the usability of UX-MAPPER needs improvement.

7.7. SUMMARY

In this chapter, we presented the results of the study conducted with the goal of investigating the acceptance, relevance, and usefulness of UX-MAPPER for practitioners. We also compared two state-of-the-art approaches that extract features from app reviews.

The results of this study suggest that UX-MAPPER helps practitioners to analyze user reviews. The organization into factors facilitate practitioners to find reviews related to a specific topic, such as bugs and improvements, which can reduce the time and effort necessary to identify such reviews. The outcomes of feature extraction also allow practitioners to verify the most requested features, which can be useful to identify users’ preferences, needs, and trends.

Finally, the combination of the features extracted with the rating analysis can provide valuable information on the impact of these features on UX. Such information can help practitioners identify which features to prioritize when planning the next release or when analyzing the reviews of concurrent apps to identify the most important features they should include when developing a new app.

In addition to these benefits and positive acceptance, we also identified the following improvement opportunities:

- Improve the usability of UX-MAPPER:
  - Make the factor's rating distribution graph more intuitive by providing legends or changing it to another form of representation;
  - Make the graphs' labels clickable and make it clearer that the user can click on the bars to visualize the reviews;
  - Make the tool not hide other star ratings after selecting one of them for filtering purposes;
  - Make the tool mobile friendly.
- Optimize the feature extraction process:
  - Regarding RE-BERT, we could employ a similarity check as we implemented in SAFE to group similar features. Then, this set would be analyzed to select the most frequent feature comprising two or more words (n-gram ≥ 2), if available. By doing so, it would be possible to present features that provide more context. Regarding SAFE, we could fine-tune the similarity check to group similar terms more effectively.

CHAPTER 8 – FINAL CONSIDERATIONS AND FUTURE WORK

This chapter presents the final remarks on our proposal, UX-MAPPER, an approach to automatically analyze app store reviews to identify factors that affect UX. We present the main findings and contributions of this work, as well as future perspectives for the research on the field of Software Engineering in terms of UX.

8.1. FINAL CONSIDERATIONS

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 to address UX issues from analyzing several self-reports of end-users' experiences in the wild. However, analyzing such reviews is costly and time-consuming, which highlights the necessity to develop approaches that automatically analyze such reviews and provide meaningful results.

We conducted this research guided by the following research 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 automatically identifying the factors that affect UX through the analysis of user reviews from app stores. To achieve this goal, we defined three specific goals as follows:

1. Provide a body of knowledge regarding different factors that can affect UX in mobile apps;
2. Define automated strategies to support the software development process by identifying the factors that lead to more positive or negative reviews;
3. Support practitioners in identifying users' most frequently reported app features that they should consider during mobile software development.

To guide the conduction of this research towards the development of an artifact, we followed the Design Science Research (DSR) methodology. First, we conducted an exploratory study to investigate the effect of a set of factors on UX. We found that some factors can influence how users perceive their experience towards more negative or positive evaluations. Such results revealed the importance of identifying these factors to the software development process. By identifying these factors, it would be possible to trace out strategies to design products that focus on factors that influences users' perceptions of UX more positively, while reducing the effect of the factors that lead to negative perceptions. To do so, we performed a systematic mapping study. We found a total of 31 factors and their effects on UX, which formed the theoretical basis of our proposal and the body of knowledge defined in the first specific goal.

After identifying these factors, we begin investigating the relevance of an automated approach to analyze app store reviews. To do so, we conducted an exploratory study with practitioners from the industry. The results revealed that they analyze such reviews as part of their jobs to improve their apps. They also had positive expectations towards an automated approach that would facilitate their tasks, revealing the relevance of our proposal.

Next, we developed an MVP of our proposal and conducted a feasibility study to assess its usefulness and acceptance from the practitioners' points of view. The results indicated that the practitioners were able to think of requirements to improve the app based on the reviews returned from our approach. We also identified some improvement opportunities to develop our artifact.

Based on the results of the systematic mapping study and these exploratory studies, we developed our artifact, called UX-MAPPER, an automated approach to analyze app store reviews and identify the factors that are affecting UX positively or negatively. To develop it, we tested different classification algorithms and automated feature extraction approaches, thus reaching our second specific goal.

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 the software development by providing a set of factors affecting UX and the most frequent features reported by users. Thus, reaching our third specific goal.

Finally, back to 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. (2016), 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 are 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 that it supports identifying the main factors that are affecting the experience.

8.2. THREATS TO VALIDITY

In this subsection, we describe the main threats to validity of this research. Regarding the tool, the performance of the classifier from the factor extraction component might have been affected by the imbalanced dataset. 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 from where the practitioners are from, in which people are willing to help each other. To minimize this bias, we told the participants to be as much as critical as possible, given that we wanted to obtain feedback to improve UX-MAPPER. The positive perception may have been influenced by their previous experience with automated text analysis approaches. To minimize this bias, we divided the participants into two groups (with and without previous experience with automated approaches) and analyzed the results accordingly.

8.3. CONTRIBUTIONS

The main contributions of this research are:

- A secondary study addressing publications that investigated factors that could affect users' perception of the experience, which implied in:
    - An overview of the state of the art on analyzing user reviews from app stores with focus on UX;
    - A set of factors that could affect users' perception of their experience with mobile applications and their effects;
    - An overview of the methods employed to analyze the reviews;

  - 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 identifying improving opportunities;
- Empirical evidence regarding the usefulness, relevance, and acceptance of using automated approaches to analyze app store reviews from practitioners' perspective;
- Dissemination of the results and the knowledge obtained during the conduction of this research through the publication in journal papers and conference proceedings.

8.4. PUBLICATIONS

This research resulted in the following publications:

- Nakamura, W. T.; de Oliveira, E. C.; de Oliveira, E. H. T.; Redmiles, D.; Conte, T. (2022). What factors affect the UX in mobile apps? A systematic mapping study on the analysis of app store review. Journal of Systems and Software. (Under review)
  - Describe the results of the systematic mapping study conducted to identify factors that affect the UX in mobile apps (CHAPTER 4).
- Nakamura, W. T.; Marques, L. C.; Redmiles, D.; de Oliveira, E. H. T.; Conte, T. (2022). Investigating the influence of different factors on the UX evaluation of a mobile application. International Journal of Human-Computer Interaction. (Under review)
  - Presents the results of the empirical study conducted to investigate the influence of number of problems, previous experience with similar apps, and interaction sequencing (CHAPTER 3).
- Nakamura, W. T., de Souza, J. C., Teixeira, L. M., Silva, A., da Silva, R., Gadelha, B., & Conte, T. (2021). Requirements Behind Reviews: How do Software Practitioners See App User Reviews to Think of Requirements?. In XX Brazilian Symposium on Software Quality (pp. 1-9).
  - Presents the results of our feasibility study in which we investigated whether the use of an automated approach to analyze app store reviews would help practitioners to think of requirements to improve an app (Section 5.3).
- Nakamura, W. T., Ahmed, I., Redmiles, D., Oliveira, E., Fernandes, D., de Oliveira, E. H., & Conte, T. (2021). Are UX Evaluation Methods Providing the Same Big Picture?. Sensors, 21(10), 3480.

  - Describe the results of an exploratory study in which we compared the results of two UX evaluation methods and investigated how UX evolves over time.
- Nakamura, W. T., de Oliveira, E. H., & Conte, T. (2019). 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 (pp. 1-6).
  - Presents the initial ad-hoc literature search to address publications reporting contradictory results, which may indicate the possibility of factors influencing UX.

We also had the following publications in collaboration:

- Marques, L., Matsubara, P. G., Nakamura, W. T., Ferreira, B. M., Wiese, I. S., Gadelha, B. F., ... & Conte, T. U. (2021). Understanding UX Better: A New Technique to Go beyond Emotion Assessment. Sensors, 21(21), 7183.
  - Describe the development and evaluation of UX-Tips, a UX evaluation method we applied in the empirical study (CHAPTER 3).
- de Souza Filho, J. C., Nakamura, W. T., Teixeira, L. M., da Silva, R. P., Gadelha, B. F., & Conte, T. U. (2021). Towards a Data-Driven Requirements Elicitation Tool through the Lens of Design Thinking. In Proceedings of the 23rd International Conference on Enterprise Information Systems (ICEIS 2021)-Volume (Vol. 2, pp. 283-290).
  - Presents the results of the exploratory study (Section 5.2) we conducted to investigate whether and how practitioners analyze app store reviews and the expectations towards an approach that automates the analysis process.
- Marques, L., Matsubara, P., Nakamura, W., Wiese, I., Zaina, L., & Conte, T. (2019). UX-Tips: A UX evaluation technique to support the identification of software application problems. In Proceedings of the XXXIII Brazilian Symposium on Software Engineering (pp. 224-233).
  - Presents the results of a study to investigate the feasibility of UX-Tips and an empirical study comparing it with another UX evaluation method.

8.5. FUTURE PERSPECTIVES

This research allowed the development of UX-MAPPER, an approach to analyzing app store reviews to identify factors that are affecting UX. From the results of our research, we identified opportunities for improving UX-MAPPER and conducting future research. Regarding UX-MAPPER, we identified the following improvement opportunities:

Inclusion of 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 on the number of users making reviews.

Functionality to compare 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 identify 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 the development of this type applications.

Improving UX-MAPPER's usability and compatibility with mobile devices: the results of our study revealed some usability problems that affected participants' interaction (Section 7.5.2). 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:

Identify the influence of cultural and gender aspects in reviews: previous works indicate that users' evaluation can be affected by culture (Guzman et al., 2018) but not gender (Guzman and Paredes Rojas, 2019). However, the small datasets from these studies raise the need for further investigation. Researchers could also investigate the influence of these factors on how users' write their reviews and express themselves. For instance, do culture and gender determine how users write their reviews (e.g., tonality, readability)? If so, how they affect machine learning models' results?

Investigate the generalizability of UX-MAPPER to other contexts: we developed and tested UX-MAPPER with reviews from Google Play Store. Further studies could investigate whether reviews written by users in other 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 to analyze reviews from other domains, such as software products in general.

REFERENCES

Akoglu, H. (2018). User’s guide to correlation coefficients. Turkish Journal of Emergency Medicine, 18(3), 91–93. https://doi.org/10.1016/j.tjem.2018.08.001

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. 2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR), 187–197. https://doi.org/10.1109/MSR.2017.42

Alhadreti, O., and Mayhew, P. (2018). Rethinking Thinking Aloud: A Comparison of Three Think-Aloud Protocols. Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI ’18, 1–12. https://doi.org/10.1145/3173574.3173618

Alves, R., Valente, P., and Nunes, N. J. (2014). The state of user experience evaluation practice. Proceedings of the 8th Nordic Conference on Human-Computer Interaction Fun, Fast, Foundational - NordiCHI ’14, 93–102. https://doi.org/10.1145/2639189.2641208

Ardito, C., Buono, P., Caivano, D., Costabile, M. F., and Lanzilotti, R. (2014). Investigating and promoting UX practice in industry: An experimental study. International Journal of Human-Computer Studies, 72(6), 542–551. https://doi.org/10.1016/j.ijhcs.2013.10.004

Bakiu, E., and Guzman, E. (2017). Which Feature is Unusable? Detecting Usability and User Experience Issues from User Reviews. 2017 IEEE 25th International Requirements Engineering Conference Workshops (REW), 182–187. https://doi.org/10.1109/REW.2017.76

Bargas-Avila, J. A., and Hornbæk, K. (2011). Old Wine in New Bottles or Novel Challenges: A Critical Analysis of Empirical Studies of User Experience. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2689–2698. https://doi.org/10.1145/1978942.1979336

Bevan, N. (2008). Classifying and selecting UX and usability measures. International Workshop on Meaningful Measures: Valid Useful User Experience Measurement, 13–18.

Bevan, N. (2009). What is the difference between the purpose of usability and user experience evaluation methods. Proceedings of the Workshop UXEM, 9, 1–4.

Biduski, D., Bellei, E. A., Rodriguez, J. P. M., Zaina, L. A. M., and De Marchi, A. C. B. (2020). Assessing long-term user experience on a mobile health application through an in-app embedded conversation-based questionnaire. Computers in Human Behavior, 104, 106169. https://doi.org/10.1016/j.chb.2019.106169

Bødker, S. (2006). When second wave HCI meets third wave challenges. Proceedings of the 4th Nordic Conference on Human-Computer Interaction Changing Roles - NordiCHI ’06, 1–8. https://doi.org/10.1145/1182475.1182476

Bødker, S. (2015). Third-wave HCI, 10 years later—Participation and sharing. Interactions, 22(5), 24–31. https://doi.org/10.1145/2804405

Bolchini, D., Garzotto, F., and Sorce, F. (2009). Does Branding Need Web Usability? A Value-Oriented Empirical Study. In T. Gross, J. Gulliksen, P. Kotzé, L. Oestreicher, P. Palanque, R. O. Prates, and M. Winckler (Eds.), Human-Computer Interaction – INTERACT 2009 (Vol. 5727, pp. 652–665). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-03658-3_70

Bopp, J. A., Mekler, E. D., and Opwis, K. (2016). Negative Emotion, Positive Experience?: Emotionally Moving Moments in Digital Games. Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, 2996–3006. https://doi.org/10.1145/2858036.2858227

Borsci, S., Federici, S., Bacci, S., Gnaldi, M., and Bartolucci, F. (2015). Assessing User Satisfaction in the Era of User Experience: Comparison of the SUS, UMUX, and UMUX-LITE as a Function of Product Experience. International Journal of Human-Computer Interaction, 31(8), 484–495. https://doi.org/10.1080/10447318.2015.1064648

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.

Bruun, A., and Ahm, S. (2015). Mind the Gap! Comparing Retrospective and Concurrent Ratings of Emotion in User Experience Evaluation. In J. Abascal, S. Barbosa, M. Fetter, T. Gross, P. Palanque, and M. Winckler (Eds.), Human-Computer Interaction – INTERACT 2015 (pp. 237–254). Springer International Publishing.

Chen, N., Lin, J., Hoi, S. C. H., Xiao, X., and Zhang, B. (2014). AR-miner: Mining informative reviews for developers from mobile app marketplace. Proceedings of the 36th International Conference on Software Engineering - ICSE 2014, 767–778. https://doi.org/10.1145/2568225.2568263

Chen, T., and Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794. https://doi.org/10.1145/2939672.2939785

Cockburn, A., Quinn, P., and Gutwin, C. (2015). Examining the Peak-End Effects of Subjective Experience. Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI ’15, 357–366. https://doi.org/10.1145/2702123.2702139

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. https://doi.org/10.1016/j.ijhcs.2017.07.005

Cohen, J. (1960). A Coefficient of Agreement for Nominal Scales. Educational and Psychological Measurement, 20(1), 37–46. https://doi.org/10.1177/001316446002000104

Cohen, J. (1988). Statistical power analysis for the behavioral sciences (2nd ed). L. Erlbaum Associates.

Cronbach, L. J. (1951). Coefficient alpha and the internal structure of tests. Psychometrika, 16(3), 297–334.

Dąbrowski, J., Letier, E., Perini, A., and Susi, A. (2020). Mining User Opinions to Support Requirement Engineering: An Empirical Study. In S. Dustdar, E. Yu, C. Salinesi, D. Rieu, and V. Pant (Eds.), Advanced Information Systems Engineering (Vol. 12127, pp. 401–416). Springer International Publishing. https://doi.org/10.1007/978-3-030-49435-3_25

Darin, T., Coelho, B., and Borges, B. (2019). Which Instrument Should I Use? Supporting Decision-Making About the Evaluation of User Experience. In A. Marcus and W. Wang (Eds.), Design, User Experience, and Usability. Practice and Case Studies (Vol. 11586, pp. 49–67). Springer International Publishing. https://doi.org/10.1007/978-3-030-23535-2_4

Davis, F. D., Bagozzi, R. P., and Warshaw, P. R. (1989). User Acceptance of Computer Technology: A Comparison of Two Theoretical Models. Management Science, 35(8), 982–1003. https://doi.org/10.1287/mnsc.35.8.982

Davis, F. D., and Venkatesh, V. (1996). A critical assessment of potential measurement biases in the technology acceptance model: Three experiments. International Journal of Human-Computer Studies, 45(1), 19–45. https://doi.org/10.1006/ijhc.1996.0040

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. Proceedings of the 17th Brazilian Symposium on Human Factors in Computing Systems - IHC 2018, 1–11. https://doi.org/10.1145/3274192.3274201

de Araújo, A. F., and Marcacini, R. M. (2021). RE-BERT: Automatic extraction of software requirements from app reviews using BERT language model. Proceedings of the 36th Annual ACM Symposium on Applied Computing, 1321–1327. https://doi.org/10.1145/3412841.3442006

de Lera, E., and Garreta-Domingo, M. (2007, September 1). Ten Emotion Heuristics: Guidelines for assessing the user’s affective dimension easily and cost-effectively. Proceedings of HCI 2007 The 21st British HCI Group Annual Conference University of Lancaster, UK. https://doi.org/10.14236/ewic/HCI2007.43

Díaz-Oreiro, López, Quesada, and Guerrero. (2019). Standardized Questionnaires for User Experience Evaluation: A Systematic Literature Review. Proceedings, 31(1), 14. https://doi.org/10.3390/proceedings2019031014

Dragoni, M., Federici, M., and Rexha, A. (2019). An unsupervised aspect extraction strategy for monitoring real-time reviews stream. Information Processing & Management, 56(3), 1103–1118. https://doi.org/10.1016/j.ipm.2018.04.010

Dresch, A., Lacerda, D. P., and Antunes Jr, J. A. V. (2015). Design Science Research: A Method for Science and Technology Advancement. Springer International Publishing. https://doi.org/10.1007/978-3-319-07374-3

Duarte, E. F., and Baranauskas, M. C. C. (2016). Revisiting the Three HCI Waves: A Preliminary Discussion on Philosophy of Science and Research Paradigms. Proceedings of the 15th Brazilian Symposium on Human Factors in Computer Systems - IHC '16, 1–4. https://doi.org/10.1145/3033701.3033740

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? Proceedings of the XXXII Brazilian Symposium on Software Engineering, 132–141. https://doi.org/10.1145/3266237.3266272

Dyba, T., Dingsoyr, T., and Hanssen, G. K. (2007). Applying Systematic Reviews to Diverse Study Types: An Experience Report. First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007), 225–234. https://doi.org/10.1109/ESEM.2007.59

Ebrahimi, F., Tushev, M., and Mahmoud, A. (2020). Mobile app privacy in software engineering research: A systematic mapping study. Information and Software Technology, 106466. https://doi.org/10.1016/j.infsof.2020.106466

Fernandez, A., Abrahão, S., and Insfran, E. (2013). Empirical validation of a usability inspection method for model-driven Web development. Journal of Systems and Software, 86(1), 161–186. https://doi.org/10.1016/j.jss.2012.07.043

Fernandez, A., Insfran, E., and Abrahão, S. (2011). Usability evaluation methods for the web: A systematic mapping study. Information and Software Technology, 53(8), 789–817. https://doi.org/10.1016/j.infsof.2011.02.007

Folmer, E., van Gurp, J., and Bosch, J. (2003). A framework for capturing the relationship between usability and software architecture. Software Process: Improvement and Practice, 8(2), 67–87. https://doi.org/10.1002/spip.171

Fritz, C. O., Morris, P. E., and Richler, J. J. (2012). Effect size estimates: Current use, calculations, and interpretation. Journal of Experimental Psychology: General, 141(1), 2–18. https://doi.org/10.1037/a0024338

Fu, B., Lin, J., Li, L., Faloutsos, C., Hong, J., and Sadeh, N. (2013). Why people hate your app: Making sense of user feedback in a mobile app store. Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '13, 1276. https://doi.org/10.1145/2487575.2488202

Fuchsberger, V., Moser, C., and Tscheligi, M. (2012). Values in action (ViA): Combining usability, user experience and user acceptance. Proceedings of the 2012 ACM Annual Conference Extended Abstracts on Human Factors in Computing Systems Extended Abstracts - CHI EA '12, 1793. https://doi.org/10.1145/2212776.2223711

Genc-Nayebi, N., and Abran, A. (2017). A systematic literature review: Opinion mining studies from mobile app store user reviews. Journal of Systems and Software, 125, 207–219.

Gomez, M., Rouvoy, R., Monperrus, M., and Seinturier, L. (2015). A Recommender System of Buggy App Checkers for App Store Moderators. 2015 2nd ACM International Conference on Mobile Software Engineering and Systems, 1–11. https://doi.org/10.1109/MobileSoft.2015.8

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. Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16, 5608–5619. 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. 2014 IEEE 22nd International Requirements Engineering Conference (RE), 153–162. https://doi.org/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. Proceedings of the 40th International Conference on Software Engineering Software Engineering in Society - ICSE-SEIS '18, 13–22. https://doi.org/10.1145/3183428.3183436

Guzman, E., and Paredes Rojas, A. (2019). Gender and user feedback: An exploratory study. In L. S.-W. Damian D. Perini A. (Ed.), Proceedings of the IEEE International Conference on Requirements Engineering (Vols. 2019-September, pp. 381–385). IEEE Computer Society. https://doi.org/10.1109/RE.2019.00049

Harman, M., Yue Jia, and Yuanyuan Zhang. (2012). App store mining and analysis: MSR for app stores. 2012 9th IEEE Working Conference on Mining Software Repositories (MSR), 108–111. https://doi.org/10.1109/MSR.2012.6224306

Hartson, H. R., Andre, T. S., and Williges, R. C. (2003). Criteria For Evaluating Usability Evaluation Methods. International Journal of Human-Computer Interaction, 15(1), 145–181. https://doi.org/10.1207/S15327590IJHC1501_13

Hassenzahl, M. (2004). The Interplay of Beauty, Goodness, and Usability in Interactive Products. Human-Computer Interaction, 19(4), 319–349. https://doi.org/10.1207/s15327051hci1904_2

Hassenzahl, M. (2007). The hedonic/pragmatic model of user experience. Towards a UX Manifesto, 10, 10–14.

Hassenzahl, M. (2018a). The Thing and I (Summer of '17 Remix). In M. Blythe and A. Monk (Eds.), Funology 2: From Usability to Enjoyment (pp. 17–31). Springer International Publishing. https://doi.org/10.1007/978-3-319-68213-6_2

Hassenzahl, M. (2018b). The Thing and I: Understanding the Relationship Between User and Product. In M. Blythe and A. Monk (Eds.), Funology 2 (pp. 301–313). Springer, Cham. https://doi.org/10.1007/978-3-319-68213-6_19

Hassenzahl, M., Burmester, M., and Koller, F. (2003). AttrakDiff: Ein Fragebogen zur Messung wahrgenommener hedonischer und pragmatischer Qualität. In Mensch & computer 2003 (pp. 187–196). Springer.

Hassenzahl, M., Diefenbach, S., and Göritz, A. (2010). Needs, affect, and interactive products – Facets of user experience. Interacting with Computers, 22(5), 353–362. https://doi.org/10.1016/j.intcom.2010.04.002

Hassenzahl, M., and Sandweg, N. (2004). From mental effort to perceived usability: Transforming experiences into summary assessments. Extended Abstracts of the 2004 Conference on Human Factors and Computing Systems - CHI ’04, 1283. https://doi.org/10.1145/985921.986044

Hassenzahl, M., and Tractinsky, N. (2006). User experience—A research agenda. Behaviour & Information Technology, 25(2), 91–97. https://doi.org/10.1080/01449290500330331

Hedegaard, S., and Simonsen, J. G. (2013). Extracting usability and user experience information from online user reviews. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI ’13, 2089. https://doi.org/10.1145/2470654.2481286

Hedegaard, S., and Simonsen, J. G. (2014). Mining until it hurts: Automatic extraction of usability issues from online reviews compared to traditional usability evaluation. Proceedings of the 8th Nordic Conference on Human-Computer Interaction Fun, Fast, Foundational - NordiCHI ’14, 157–166. https://doi.org/10.1145/2639189.2639211

Hevner, A. R. (2007). A Three Cycle View of Design Science Research. Scandinavian Journal of Information Systems, 19(2), 87–92.

Hevner, A. R., and Chatterjee, S. (2010). Design Research in Information Systems (Vol. 22). Springer US. https://doi.org/10.1007/978-1-4419-5653-8

Hoon, L., Vasa, R., Schneider, J.-G., and Mouzakis, K. (2012). A preliminary analysis of vocabulary in mobile app user reviews. Proceedings of the 24th Australian Computer-Human Interaction Conference on - OzCHI ’12, 245–248. https://doi.org/10.1145/2414536.2414578

Hu, H., Bezemer, C.-P., and Hassan, A. E. (2018). Studying the consistency of star ratings and the complaints in 1 & 2-star user reviews for top free cross-platform Android and iOS apps. Empirical Software Engineering, 23(6), 3442–3475. https://doi.org/10.1007/s10664-018-9604-y

ISO 9241-11. (2018). Ergonomics of human-system interaction—Part 11: Usability: Definitions and concepts.

ISO/IEC 25012. (2008). Software engineering—Software product Quality Requirements and Evaluation (SQuaRE)—Data quality model. https://www.iso.org/standard/35736.html

Jang, J., and Yi, M. Y. (2017). Modeling User Satisfaction from the Extraction of User Experience Elements in Online Product Reviews. Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems - CHI EA '17, 1718–1725. 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. 2017 IEEE 25th International Requirements Engineering Conference (RE), 21–30. https://doi.org/10.1109/RE.2017.71

Johannessen, G. H. J., and Hornbæk, K. (2014). Must evaluation methods be about usability? Devising and assessing the utility inspection method. Behaviour & Information Technology, 33(2), 195–206. https://doi.org/10.1080/0144929X.2012.751708

Ketola, P., and Roto, V. (2009). On User Experience Measurement Needs: 12.

Kim, H. K., Han, S. H., Park, J., and Park, W. (2015). How User Experience Changes over Time: A Case Study of Social Network Services: How User Experience Changes over Time. Human Factors and Ergonomics in Manufacturing & Service Industries, 25(6), 659–673. https://doi.org/10.1002/hfm.20583

Kitchenham, B. A., Budgen, D., and Brereton, P. (2015). Evidence-based software engineering and systematic reviews (Vol. 4). CRC press.

Kitchenham, B., and Charters, S. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering.

Kou, Y., and Gray, C. M. (2019). A Practice-Led Account of the Conceptual Evolution of UX Knowledge. Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19, 1–13. https://doi.org/10.1145/3290605.3300279

Kujala, S., and Miron-Shatz, T. (2015). The evolving role of expectations in long-term user experience. Proceedings of the 19th International Academic Mindtrek Conference on - AcademicMindTrek '15, 167–174. https://doi.org/10.1145/2818187.2818271

Kujala, S., Mugge, R., and Miron-Shatz, T. (2017). The role of expectations in service evaluation: A longitudinal study of a proximity mobile payment service. International Journal of Human-Computer Studies, 98, 51–61. https://doi.org/10.1016/j.ijhcs.2016.09.011

Lallemand, C., Gronier, G., and Koenig, V. (2015). User experience: A concept without consensus? Exploring practitioners' perspectives through an international survey. Computers in Human Behavior, 43, 35–48. https://doi.org/10.1016/j.chb.2014.10.048

Lallemand, C., and Koenig, V. (2017). How Could an Intranet be Like a Friend to Me?: Why Standardized UX Scales Don't Always Fit. Proceedings of the European Conference on Cognitive Ergonomics 2017 - ECCE 2017, 9–16. https://doi.org/10.1145/3121283.3121288

Lallemand, C., Koenig, V., and Gronier, G. (2014). How relevant is an expert evaluation of user experience based on a psychological needs-driven approach? Proceedings of the

8th Nordic Conference on Human-Computer Interaction Fun, Fast, Foundational - NordiCHI ’14, 11–20. https://doi.org/10.1145/2639189.2639214

Landis, J. R., and Koch, G. G. (1977). The Measurement of Observer Agreement for Categorical Data. Biometrics, 33(1), 159. https://doi.org/10.2307/2529310

Laugwitz, B., Held, T., and Schrepp, M. (2008). Construction and Evaluation of a User Experience Questionnaire. In A. Holzinger (Ed.), HCI and Usability for Education and Work (Vol. 5298, pp. 63–76). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-89350-9_6

Law, E. L.-C., and Hvannberg, E. T. (2002). Complementarity and Convergence of Heuristic Evaluation and Usability Test: A Case Study of UNIVERSAL Brokerage Platform. Proceedings of the Second Nordic Conference on Human-Computer Interaction, 71–80.

Law, E. L.-C., Roto, V., Hassenzahl, M., Vermeeren, A. P. O. S., and Kort, J. (2009). Understanding, Scoping and Defining User Experience: A Survey Approach. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 719–728. https://doi.org/10.1145/1518701.1518813

Law, E. L.-C., and van Schaik, P. (2010). Modelling user experience – An agenda for research and practice. Interacting with Computers, 22(5), 313–322. https://doi.org/10.1016/j.intcom.2010.04.006

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. https://doi.org/10.1016/j.ijhcs.2013.09.006

Lu, M., and Liang, P. (2017). Automatic Classification of Non-Functional Requirements from Augmented App User Reviews. Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering, 344–353. https://doi.org/10.1145/3084226.3084241

Luiz, W., Viegas, F., Alencar, R., Mourão, F., Salles, T., Carvalho, D., Gonçalves, M. A., and Rocha, L. (2018). A Feature-Oriented Sentiment Rating for Mobile App Reviews. Proceedings of the 2018 World Wide Web Conference, 1909–1918. https://doi.org/10.1145/3178876.3186168

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

Maalej, W., and Nabil, H. (2015). Bug report, feature request, or simply praise? On automatically classifying app reviews. 2015 IEEE 23rd International Requirements Engineering Conference (RE), 116–125. https://doi.org/10.1109/RE.2015.7320414

Maguire, M., and Isherwood, P. (2018). A Comparison of User Testing and Heuristic Evaluation Methods for Identifying Website Usability Problems. In A. Marcus and W. Wang (Eds.), Design, User Experience, and Usability: Theory and Practice (Vol. 10918, pp. 429–438). Springer International Publishing. https://doi.org/10.1007/978-3-319-91797-9_31

Marques, L. C. (2019). UX-Tips: Uma Técnica de Avaliação de User eXperience para Aplicações de Software [Universidade Federal do Amazonas (UFAM)]. https://tede.ufam.edu.br/handle/tede/6984

Marques, L. C., Nakamura, W. T., Valentim, N., Rivero, L., and Conte, T. (2018). Do Scale Type Techniques Identify Problems that Affect User eXperience? User Experience Evaluation of a Mobile Application (S). The 30th International Conference on Software Engineering and Knowledge Engineering, 451–501. https://doi.org/10.18293/SEKE2018-161

Martens, D., and Johann, T. (2017). On the emotion of users in app reviews. Proceedings - 2017 IEEE/ACM 2nd International Workshop on Emotion Awareness in Software Engineering, SEmotion 2017, 8–14. https://doi.org/10.1109/SEmotion.2017.6

Martens, D., and Maalej, W. (2019). Release Early, Release Often, and Watch Your Users' Emotions: Lessons From Emotional Patterns. IEEE Software, 36(5), 32–37. https://doi.org/10.1109/MS.2019.2923603

Masrury, R. A., Fannisa, and Alamsyah, A. (2019). Analyzing Tourism Mobile Applications Perceived Quality using Sentiment Analysis and Topic Modeling. 2019 7th International Conference on Information and Communication Technology (ICoICT), 1–6. https://doi.org/10.1109/ICoICT.2019.8835255

Matera, M., Costabile, M. F., Garzotto, F., and Paolini, P. (2002). SUE inspection: An effective method for systematic usability evaluation of hypermedia. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 32(1), 93–103. https://doi.org/10.1109/3468.995532

McIlroy, S., Ali, N., Khalid, H., and E. Hassan, A. (2016). Analyzing and automatically labelling the types of user issues that are raised in mobile app reviews. Empirical Software Engineering, 21(3), 1067–1106. https://doi.org/10.1007/s10664-015-9375-7

Mendes, E., Wohlin, C., Felizardo, K., and Kalinowski, M. (2020). When to update systematic literature reviews in software engineering. Journal of Systems and Software, 167, 110607. https://doi.org/10.1016/j.jss.2020.110607

Metatla, O., Correia, N. N., Martin, F., Bryan-Kinns, N., and Stockman, T. (2016). Tap the ShapeTones: Exploring the Effects of Crossmodal Congruence in an Audio-Visual Interface. Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, 1055–1066. https://doi.org/10.1145/2858036.2858456

Miron-Shatz, T., Stone, A., and Kahneman, D. (2009). Memories of yesterday's emotions: Does the valence of experience affect the memory-experience gap? Emotion, 9(6), 885–891. https://doi.org/10.1037/a0017823

Moreno, A. M., Seffah, A., Capilla, R., and Sanchez-Segura, M.-I. (2013). HCI Practices for Building Usable Software. Computer, 46(4), 100–102. https://doi.org/10.1109/MC.2013.133

Nakamura, W. T., de Oliveira, E. H. T., and Conte, T. (2019). Negative Emotions, Positive Experience: What Are We Doing Wrong When Evaluating the UX? Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems - CHI EA '19, 1–6. https://doi.org/10.1145/3290607.3313000

Nakamura, W. T., Marques, L. C., Ferreira, B. M., Barbosa, S. D. J., and Conte, T. (2020). To Inspect or to Test? What Approach Provides Better Results When it comes to usability and UX? Proceedings of the 22nd International Conference on Enterprise Information Systems - Volume 2: ICEIS, 487–498. https://doi.org/10.5220/0009367904870498

Nakamura, W. T., Marques, L. C., Rivero, L., De Oliveira, E. H. T., and Conte, T. (2019). Are scale-based techniques enough for learners to convey their UX when using a Learning Management System? Revista Brasileira de Informática Na Educação, 27(01), 104. https://doi.org/10.5753/rbie.2019.27.01.104

Nalepa, J., and Kawulok, M. (2019). Selecting training sets for support vector machines: A review. Artificial Intelligence Review, 52(2), 857–900. https://doi.org/10.1007/s10462-017-9611-1

Nayebi, M., Cho, H., and Ruhe, G. (2018). App Store Mining is Not Enough for App Improvement. Empirical Softw. Engg., 23(5), 2764–2794. https://doi.org/10.1007/s10664-018-9601-1

Norman, D., Miller, J., and Henderson, A. (1995). What you see, some of what's in the future, and how we go about doing it: HI at Apple Computer. Conference Companion on Human Factors in Computing Systems - CHI '95, 155. https://doi.org/10.1145/223355.223477

Oztekin, A., Delen, D., Turkyilmaz, A., and Zaim, S. (2013). A machine learning-based usability evaluation method for eLearning systems. Decision Support Systems, 56, 63–73. https://doi.org/10.1016/j.dss.2013.05.003

Pagano, D., and Maalej, W. (2013). User feedback in the appstore: An empirical study. 2013 21st IEEE International Requirements Engineering Conference (RE), 125–134. https://doi.org/10.1109/RE.2013.6636712

Palomba, F., Linares-Vásquez, M., Bavota, G., Oliveto, R., Penta, M. D., Poshyvanyk, D., and Lucia, A. D. (2018). Crowdsourcing user reviews to support the evolution of mobile apps. Journal of Systems and Software, 137, 143–162. https://doi.org/10.1016/j.jss.2017.11.043

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. Proceedings of the 39th International Conference on Software Engineering, 106–117.

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. 2015 IEEE International Conference on Software Maintenance and Evolution (ICSME), 281–290. https://doi.org/10.1109/ICSM.2015.7332474

Panosian, H. (2017). Submitting Your App. In Learn iOS Application Distribution (pp. 153–158). Springer.

Pennebaker, J. W., Boyd, R. L., Jordan, K., and Blackburn, K. (2015). The development and psychometric properties of LIWC2015. University of Texas at Austin.

Petersen, K., Vakkalanka, S., and Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. Information and Software Technology, 64, 1–18. https://doi.org/10.1016/j.infsof.2015.03.007

Pettersson, I., Lachner, F., Frison, A.-K., Riener, A., and Butz, A. (2018). A Bermuda Triangle?: A Review of Method Application and Triangulation in User Experience Evaluation. Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, 461:1-461:16. https://doi.org/10.1145/3173574.3174035

Pine, B. J., Pine, J., and Gilmore, J. H. (1999). The experience economy: Work is theatre & every business a stage. Harvard Business Press.

Reimers, N., and Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. ArXiv:1908.10084 [Cs]. http://arxiv.org/abs/1908.10084

Rennie, J. D. M., Shih, L., Teevan, J., and Karger, D. R. (2003). Tackling the Poor Assumptions of Naive Bayes Text Classifiers. Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), 3, 616–623.

Rivero, L., and Conte, T. (2017). A Systematic Mapping Study on Research Contributions on UX Evaluation Technologies. Proceedings of the XVI Brazilian Symposium on Human Factors in Computing Systems - IHC 2017, 1–10. https://doi.org/10.1145/3160504.3160512

Rodrigues, P., Silva, I. S., Barbosa, G. A. R., Coutinho, F. R. dos S., and Mourão, F. (2017). Beyond the Stars: Towards a Novel Sentiment Rating to Evaluate Applications in Web Stores of Mobile Apps. Proceedings of the 26th International Conference on World Wide Web Companion, 109–117. https://doi.org/10.1145/3041021.3054139

Roto, V., and Lund, A. (2013). On Top of the User Experience Wave – How is Our Work Changing? CHI’13 Extended Abstracts on Human Factors in Computing Systems, 2521–2524.

Roto, V., Vermeeren, A., Väänänen-Vainio-Mattila, K., and Law, E. (2011). User Experience Evaluation – Which Method to Choose? In P. Campos, N. Graham, J. Jorge, N. Nunes, P. Palanque, and M. Winckler (Eds.), Human-Computer Interaction – INTERACT 2011 (Vol. 6949, pp. 714–715). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-23768-3_129

Sagnier, C., Loup-Escande, E., and Valléry, G. (2020). Effects of Gender and Prior Experience in Immersive User Experience with Virtual Reality. In T. Ahram and C. Falcão (Eds.), Advances in Usability and User Experience (Vol. 972, pp. 305–314). Springer International Publishing. https://doi.org/10.1007/978-3-030-19135-1_30

Santoso, H. B., Schrepp, M., Isal, R., Utomo, A. Y., and Priyogi, B. (2016). Measuring user experience of the student-centered e-learning environment. Journal of Educators Online, 13(1), 58–79.

Schmitt, B. (1999). Experiential marketing. Journal of Marketing Management, 15(1–3), 53–67.

Schneidermeier, T., Hertlein, F., and Wolff, C. (2014). Changing Paradigm – Changing Experience?: Comparative Usability Evaluation of Windows 7 and Windows 8. In A. Marcus (Ed.), Design, User Experience, and Usability. Theories, Methods, and Tools for Designing the User Experience (Vol. 8517, pp. 371–382). Springer International Publishing. https://doi.org/10.1007/978-3-319-07668-3_36

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. 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 D. Gunopulos, T. Hofmann, D. Malerba, and M. Vazirgiannis (Eds.), Machine Learning and Knowledge Discovery in Databases (Vol. 6913, pp. 145–158). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-23808-6_10

Seffah, A., Donyaee, M., Kline, R. B., and Padda, H. K. (2006). Usability measurement and metrics: A consolidated model. Software Quality Journal, 14(2), 159–178. https://doi.org/10.1007/s11219-006-7600-8

Shapiro, S. S., and Francia, R. S. (1972). An Approximate Analysis of Variance Test for Normality. Journal of the American Statistical Association, 67(337), 215–216. https://doi.org/10.1080/01621459.1972.10481232

Shteingart, H., Neiman, T., and Loewenstein, Y. (2013). The role of first impression in operant learning. Journal of Experimental Psychology: General, 142(2), 476–488. https://doi.org/10.1037/a0029550

Simmons, A., and Hoon, L. (2016). Agree to Disagree: On Labelling Helpful App Reviews. Proceedings of the 28th Australian Conference on Computer-Human Interaction, 416–420. https://doi.org/10.1145/3010915.3010976

Soleimani, S., and Law, E. L.-C. (2017). What Can Self-Reports and Acoustic Data Analyses on Emotions Tell Us? Proceedings of the 2017 Conference on Designing Interactive Systems - DIS ’17, 489–501. https://doi.org/10.1145/3064663.3064770

Tan, W., Liu, D., and Bishu, R. (2009). Web evaluation: Heuristic evaluation vs. user testing. International Journal of Industrial Ergonomics, 39(4), 621–627. https://doi.org/10.1016/j.ergon.2008.02.012

Tarafdar, M. and others. (2005). Analyzing the influence of web site design parameters on web site usability. Information Resources Management Journal (IRMJ), 18(4), 62–80.

Väänänen-Vainio-Mattila, K., Roto, V., and Hassenzahl, M. (2008). Towards Practical User Experience Evaluation Methods. Meaningful Measures: Valid Useful User Experience Measurement (VUUM), 19–22.

Venkatesh, V., and Bala, H. (2008). Technology Acceptance Model 3 and a Research Agenda on Interventions. Decision Sciences, 39(2), 273–315. https://doi.org/10.1111/j.1540-5915.2008.00192.x

Vermeeren, A. P. O. S., Law, E. L.-C., Roto, V., Obrist, M., Hoonhout, J., and Väänänen-Vainio-Mattila, K. (2010). User Experience Evaluation Methods: Current State and Development Needs. Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries, 521–530. https://doi.org/10.1145/1868914.1868973

Wang, A. I., and Tahir, R. (2020). The effect of using Kahoot! For learning – A literature review. Computers & Education, 149, 103818. https://doi.org/10.1016/j.compedu.2020.103818

Wang, H., Lu, Y., and Zhai, C. (2010). Latent Aspect Rating Analysis on Review Text Data: A Rating Regression Approach. Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 783–792.

Weichbroth, P. (2020). Usability of mobile applications: A systematic literature study. IEEE Access, 1–1. https://doi.org/10.1109/ACCESS.2020.2981892

Weichbroth, P., and Baj-Rogowska, A. (2019). Do Online Reviews Reveal Mobile Application Usability and User Experience? The Case of WhatsApp. 2019 Federated Conference on Computer Science and Information Systems (FedCSIS), 747–754. https://doi.org/10.15439/2019F289

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

Williams, B., Onsman, A., and Brown, T. (2010). Exploratory factor analysis: A five-step guide for novices. Australasian Journal of Paramedicine, 8(3). https://doi.org/10.33151/ajp.8.3.93

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. (2012). Experimentation in software engineering. Springer Science & Business Media.

Zhang, H., Babar, M. A., and Tell, P. (2011). Identifying relevant studies in software engineering. Information and Software Technology, 53(6), 625–637. https://doi.org/10.1016/j.infsof.2010.12.010

Zhang, M.-L., and Zhou, Z.-H. (2014). A Review on Multi-Label Learning Algorithms. IEEE Transactions on Knowledge and Data Engineering, 26(8), 1819–1837. https://doi.org/10.1109/TKDE.2013.39

APPENDIX A – LIST OF SELECTED PUBLICATIONS IN THE SYSTEMATIC STUDY

This appendix contains the list of all publications selected in the 2nd filter of the Systematic Mapping presented in Chapter 5.

ID	Title
S01	Guzman, E., & Rojas, A. P. (2019, September). Gender and user feedback: An exploratory study. In 2019 IEEE 27th International Requirements Engineering Conference (RE) (pp. 381-385). IEEE.
S02	Nicolai, M., Pascarella, L., Palomba, F., & Bacchelli, A. (2019, August). Healthcare Android apps: a tale of the customers' perspective. In Proceedings of the 3rd ACM SIGSOFT International Workshop on App Market Analytics (pp. 33-39).
S03	Durelli, V. H., Durelli, R. S., Endo, A. T., Cirilo, E., Luiz, W., & Rocha, L. (2018, September). Please please me: does the presence of test cases influence mobile app users' satisfaction?. In Proceedings of the XXXII Brazilian Symposium on Software Engineering (pp. 132-141).
S04	Li, X., Zhang, Z., & Stefanidis, K. (2018, September). Mobile App Evolution Analysis Based on User Reviews. In SoMeT (pp. 773-786).
S05	Martens, D., & Johann, T. (2017, May). On the emotion of users in app reviews. In 2017 IEEE/ACM 2nd International Workshop on Emotion Awareness in Software Engineering (SEmotion) (pp. 8-14). IEEE.
S06	Genc-Nayebi, N., & Abran, A. (2017). A systematic literature review: Opinion mining studies from mobile app store user reviews. Journal of Systems and Software, 125, 207-219.
S07	Bano, M., Zowghi, D., & Kearney, M. (2017, July). Feature based sentiment analysis for evaluating the mobile pedagogical affordances of apps. In IFIP World Conference on Computers in Education (pp. 281-291). Springer, Cham.
S08	Shah, F. A., Sabanin, Y., & Pfahl, D. (2016, November). Feature-based evaluation of competing apps. In Proceedings of the International Workshop on App Market Analytics (pp. 15-21).
S09	Khalid, H., Nagappan, M., & Hassan, A. E. (2015). Examining the relationship between FindBugs warnings and app ratings. Ieee Software, 33(4), 34-39.
S10	Mohan, L., Mathur, N., & Reddy, Y. R. (2015, April). Improving Mobile Banking Usability Based on Sentiments. In International Conference on Evaluation of Novel Approaches to Software Engineering (pp. 180-194). Springer, Cham.
S11	Guzman, E., & Maalej, W. (2014, August). How do users like this feature? a fine grained sentiment analysis of app reviews. In 2014 IEEE 22nd international requirements engineering conference (RE) (pp. 153-162). IEEE.
S12	Kang, D., & Park, Y. (2014). based measurement of customer satisfaction in mobile service: Sentiment analysis and VIKOR approach. Expert Systems with Applications, 41(4), 1041-1050.
S13	Fu, B., Lin, J., Li, L., Faloutsos, C., Hong, J., & Sadeh, N. (2013, August). Why people hate your app: Making sense of user feedback in a mobile app store. In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 1276-1284).

ID	Title
S14	Ha, E., & Wagner, D. (2013, January). Do android users write about electric sheep? examining consumer reviews in google play. In 2013 IEEE 10th Consumer Communications and Networking Conference (CCNC) (pp. 149-157). IEEE.
S15	Goul, M., Marjanovic, O., Baxley, S., & Vizecky, K. (2012, January). Managing the enterprise business intelligence app store: Sentiment analysis supported requirements engineering. In 2012 45th Hawaii International Conference on System Sciences (pp. 4168-4177). IEEE.
S16	Pagano, D., & Maalej, W. (2013, July). User feedback in the appstore: An empirical study. In 2013 21st IEEE international requirements engineering conference (RE) (pp. 125-134). IEEE.
S17	Luiz, W., Viegas, F., Alencar, R., Mourão, F., Salles, T., Carvalho, D., ... & Rocha, L. (2018, April). A feature-oriented sentiment rating for mobile app reviews. In Proceedings of the 2018 World Wide Web Conference (pp. 1909-1918).
S18	Harman, M., Jia, Y., & Zhang, Y. (2012, June). App store mining and analysis: MSR for app stores. In 2012 9th IEEE working conference on mining software repositories (MSR) (pp. 108-111). IEEE.
S19	Khalid, H., Shihab, E., Nagappan, M., & Hassan, A. E. (2014). What do mobile app users complain about?. IEEE software, 32(3), 70-77.
S20	Vu, P. M., Nguyen, T. T., Pham, H. V., & Nguyen, T. T. (2015, November). Mining user opinions in mobile app reviews: A keyword-based approach (t). In 2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE) (pp. 749-759). IEEE.
S21	Iacob, C., Veerappa, V., & Harrison, R. (2013, September). What are you complaining about?: a study of online reviews of mobile applications. In 27th International BCS Human Computer Interaction Conference (HCI 2013) 27 (pp. 1-6).
S22	Khalid, H., Nagappan, M., Shihab, E., & Hassan, A. E. (2014, November). Prioritizing the devices to test your app on: A case study of android game apps. In Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering (pp. 610-620).
S23	Keertipati, S., Savarimuthu, B. T. R., & Licorish, S. A. (2016, June). Approaches for prioritizing feature improvements extracted from app reviews. In Proceedings of the 20th international conference on evaluation and assessment in software engineering (pp. 1-6).
S24	Guzman, E., Oliveira, L., Steiner, Y., Wagner, L. C., & Glinz, M. (2018, May). User feedback in the app store: a cross-cultural study. In 2018 IEEE/ACM 40th International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS) (pp. 13-22). IEEE.
S25	Palomba, F., Linares-Vásquez, M., Bavota, G., Oliveto, R., Di Penta, M., Poshyvanyk, D., & De Lucia, A. (2018). Crowdsourcing user reviews to support the evolution of mobile apps. Journal of Systems and Software, 137, 143-162.

APPENDIX B – PRIMARY STUDIES MAPPING

This appendix presents the mapping of all publications with the respective answers according to each research sub-question of the systematic mapping study.

	Venue			Sample		SQ1				SQ2				SQ3						SQ4		SQ5			SQ6			SQ7		
ID	a	b	c	Apps	Reviews	a	b	c	d	a	b	c	d	a	b	c	d	e	f	a	b	a	b	c	a	b	c	a	b	Score
S01	X			9	5,036		X			X	X				X	X					X		X				X	X		3
S02	X			32,108	-				X			X				X					X		X	X			X	X		1.5
S03	X			171,000	13,000,000+	X				X	X	X		X		X				X			X			X		X		3
S04	X			59	556	X				X	X					X	X	X		X				X	X	X	X	X		3
S05	X			1,100	1,126,453		X				X	X				X	X	X		X				X	X	X	X	X		3
S06	X			161	3,279	X				X	X	X				X		X		X				X	X	X	X	X		1.5
S07	X			7	32,210	X	X			X	X	X		X	X						X	X			X	X			X	3
S08		X		8	1,487				X		X								X	X			X			X			X	3
S09	X			+99	206,751	X				X	X		X			X	X				X			X		X		X		3
S10		X		20	6,390		X			X	X						X	X		X				X		X		X		1.5
S11	X			95	2,106,605	X				X	X								X	X				X		X			X	3
S12			X	25	100.000+		X				X				X						X	X			X		X		X	3
S13		X		5,000	~2,500,000	X				X	X		X				X		X	X				X		X		X		3
S14	X			51	303,694	X					X				X						X		X		X				X	1.5
S15	X			1	4,442	X				X	X					X					X	X				X		X		3
S16			X	245	7,396,551		X			X	X	X			X	X				X				X	X	X	X	X		3
S18	X			10	25,035	X	X			X	X				X					X			X		X				X	1.5
S19		X		100	5,792	X				X	X		X			X	X				X			X	X			X		3
S20	X			60	21.000+	X				X	X			X	X	X	X				X		X			X	X	X		3
S21	X			1	1,148,032	X					X	X		X	X				X		X	X			X	X			X	1
S22	X			7	22,815	X				X	X			X	X						X	X			X	X			X	2
S23	X			7	59,204		X			X	X					X	X			X				X	X			X		3
S24	X			7	919		X			X	X						X	X			X			X			X	X		3
S25			X	8,431	383,758	X				X	X				X			X		X			X		X	X	X	X		2.5

APPENDIX C – FACTORS’ CONSOLIDATION AND POLARITY MAPPING

This appendix presents the list of all original factors identified in our systematic mapping study ordered by the name of the factor they were consolidated with. It also presents their associated polarity, the data organization, and the ID of the publication where they were extracted.

Original Factor	Consolidated Factor	Pos	Neg	Neu	Ind	Gro	Gen	Publication
Accuracy	Accuracy		X			X		S03
Adjective (negative)	Attractiveness		X				X	S04
Adjective (positive)	Attractiveness	X					X	S04
Ads in the paid version	Spam/Ads		X		X			S22
Aesthetics (negative)	Interface		X				X	S04
Aesthetics (positive)	Interface			X			X	S04
App crashing	Bugs/Crash		X				X	S10
App improvement from reviews	Update	X					X	S19
App interface	Interface	X			X			S22
App redesign	Update		X				X	S16
App version	App version			X		X		S01
Attractiveness	Attractiveness		X			X		S03
Authentication	Feature/Functionality		X				X	S11
Battery drain	Resource use		X			X		S20
Bug report	Bugs/Crash		X				X	S16
Bugs	Bugs/Crash		X				X	S05
Bugs	Bugs/Crash		X		X			S22
Calourie counter, workout tracker	Feature/Functionality	X			X			S12
Changing app requirements	Update		X				X	S16
Cheaper app	Cost			X			X	S06
Compatibility	Compatibility		X			X		S03

Original Factor	Consolidated Factor	Pos	Neg	Neu	Ind	Gro	Gen	Publication
Compatibility	Compatibility		X				X	S10
Complaints	Attractiveness		X			X		S25
Compliments	Attractiveness	X				X		S25
Connection	Network problems		X				X	S11
Connectivity	Network problems		X			X		S03
Connectivity	Network problems		X		X			S22
Content request	Improvement request			X			X	S05
Cost	Cost		X			X		S03
Cost (negative)	Cost		X				X	S04
Cost (positive)	Cost			X			X	S04
Culture	Culture	X					X	S23
Customer support (positive)	Customer support	X					X	S06
Date/Time	Date/Time			X		X		S01
Design	Interface		X		X			S22
Device model	Device		X				X	S09
Dispraise	Attractiveness		X				X	S05
Dissuasion	Recommendation		X				X	S05
Doesn't work-TR	Bugs/Crash		X				X	S04
Easiness to use	Usability	X			X			S07
Ethical aspects (misleading app description)	Misleading app description		X				X	S21
Extended time (time, long, slow)	Performance		X		X			S22
Feature removal	Feature removal		X				X	S10
Feature request	Improvement request		X				X	S05
Feature request	Improvement request		X				X	S10
Feature request	Improvement request			X		X		S25
Feature/Functionality (negative)	Feature/Functionality		X				X	S04
Feature/Functionality (positive)	Feature/Functionality	X					X	S04
Features	Feature/Functionality		X		X			S15

Original Factor	Consolidated Factor	Pos	Neg	Neu	Ind	Gro	Gen	Publication
FindBugs warnings	FindBugs warnings		X				X	S13
Functional error	Bugs/Crash		X				X	S10
Gender	Gender			X			X	S24
Helpfulness	Helpfulness	X					X	S05
Hidden cost	Cost		X				X	S10
Improvement request	Improvement request		X				X	S05
Interface design	Interface		X				X	S10
Interface style	Interface	X			X			S22
Major bugs	Bugs/Crash		X				X	S06
Media	Feature/Functionality		X			X		S03
Messaging	Feature/Functionality		X				X	S11
Minor bugs	Bugs/Crash	X					X	S06
Missing Feature/Functionalty	Improvement request			X			X	S04
Network problem	Network problems		X				X	S10
Other app	Comparison		X				X	S05
Personalization	Personalization	X				X		S18
Photo editing	Feature/Functionality	X			X			S22
Picture	Feature/Functionality		X			X		S03
Pin thing, find thing, view file, open file	Feature/Functionality	X			X			S07
Praise	Attractiveness	X					X	S05
Presence of test cases	Presence of test cases			X		X		S20
Price X Downloads / Price X Ratings	Cost			X		X	X	S02
Price/Cost	Cost			X			X	S16
Privacy and Ethical	Privacy and Ethical		X				X	S10
Problem discovering	Bugs/Crash		X			X		S25
Problem reporting	Bugs/Crash		X			X		S25
Problems after updates	Update		X			X		S20
Problems after updates	Update		X				X	S10
Problems after updates	Update		X				X	S06

Original Factor	Consolidated Factor	Pos	Neg	Neu	Ind	Gro	Gen	Publication
Promise	Improvement request		X				X	S05
Recommendation	Recommendation	X					X	S05
Request for features regarding options fo contact and status.	Improvement request		X		X			S21
Resource heavy	Resource use		X				X	S10
Search: Ease of searching of various information | Touch: Ease of clicking and dragging of various contents	Feature/Functionality		X			X		S08
Shortcoming	Feature removal		X				X	S05
Show pin, search something, update time, want upload, take photo	Feature/Functionality		X		X			S07
Simplicity, friendly, ease of use	Usability	X				X		S14
Songs	Feature/Functionality	X			X			S22
Spam/Ads	Spam/Ads		X			X		S03
Stability	Bugs/Crash		X			X		S03
Storage	Resource use		X		X			S22
Support for loading large videos	Feature/Functionality	X			X			S22
Telephony	Feature/Functionality		X			X		S03
Theme upgrades (update, stickers, themes, wallpaper)	Improvement request		X		X			S22
Themes	Feature/Functionality	X			X			S22
Time battery	Resource use	X			X			S22
Track calorie, track weight, exercise activity	Feature/Functionality			X	X			S12
Transfer of money, ability to make card payments, getting account summary, ease of access, etc.	Feature/Functionality	X				X		S14
Uninteresting content	Attractiveness		X				X	S10
Unrecoverable error	Bugs/Crash		X				X	S11
Unresponsive app	Performance		X				X	S10
Update failures	Update		X		X			S22
Updates (positive)	Update	X					X	S16
Use in tablets	Compatibility		X		X			S22

Original Factor	Consolidated Factor	Pos	Neg	Neu	Ind	Gro	Gen	Publication
User profile of an app type	User profile of an app type	X				X		S25
Versioning	Update	X					X	S06
Video and voice call quality of the app	Feature/Functionality	X			X			S21

APPENDIX D – EXTRACTION FORM FOR PRIMARY STUDIES

This appendix presents the extraction form employed to extract the information needed for answering the research question and sub-questions of the systematic mapping study.

TITLE: **AUTHORS:** **PUBLISHED IN:** **VENUE:** ( ) Conference ( ) Journal ( ) Workshop **YEAR:**	
TABLE FOR DATA EXTRACTION	
Publication Summary	Overview of the publication (What is the goal of the publication? What is the motivation?)
RESEARCH SUB-QUESTIONS	**ANSWERS**
Q1. From what source were the reviews obtained?	a) Google Play Store b) Apple App Store c) Windows Phone Store d) Other
Q2. Which information was extracted from the source?	a) Rating b) User review c) App information d) Other (specify)
Q3. Which methods were used for analyzing the data extracted?	a) Topic modeling b) Sentiment analysis c) Descriptive statistics d) Statistical tests e) Manual analysis f) Other (specify)
Q4. Was the information categorized? How?	a) Yes (describe) b) No
Q5. How was the data organized during the analysis?	a) Individual b) Group c) General
Q6. What polarity is the factor associated to?	a) Positive b) Negative c) Neutral
Q7. Was the influence of the factor on user rating or sentiment analyzed?	a) Yes b) No
Quality Assessment Questionnaire	
a) How the association of the factors to their respective polarities were presented? ( ) **Textual description** (only described the results without providing any data such as percentages, median, mean, frequency, or graphs).	

(   ) **Descriptive data** (provided information such as percentages, median, mean, frequency, or graphs).

(   ) **Statistical analysis** (performed statistical tests or correlation/regression analysis).

b) The study clearly stated the impact of the factor on evaluations rather than just presenting the factor and the polarity of the reviews associated to it, i.e., only indicating that a given factor was evaluated positively, negatively or neutrally.

( ) Disagree   ( ) Partially agree   ( ) Agree

APPENDIX E – EXTRACTION FORM FOR SECONDARY STUDIES

This appendix presents the extraction form employed to extract data from selected secondary studies returned in the systematic mapping study.

TITLE:
AUTHORS:
PUBLISHED IN:
VENUE: (   ) Conference   (   ) Journal   (   ) Workshop
YEAR:

EXTRACTION TABLE FOR SYSTEMATIC REVIEWS AND MAPPINGS	
Q1. What is the purpose of the research?	Description of the research goals.
Q2. What are the research questions?	Description of the research questions that the SRL/SML sought to answer.
Q3. Which string was used?	String used in the search.
Q4. In what fields has the string been searched?	Description of the fields in which the string was searched, such as title, abstract, or full-text.
Q5. Which databases have been queried?	Listing of the databases in which the string was run.
Q6. What are the inclusion criteria?	Description of the inclusion criteria.
Q7. How many articles are included?	Total number of articles included after the 2nd filter.
Q8. What information is extracted from the articles?	Description of the fields used in the extraction form.
Q9. Describe the analysis of results.	Description of how the analysis was performed and its results.
Q10. What are the limitations of this SRL/SML?	Description of SRL/SML limitations.

ANNEX A – ATTRAKDIFF

This appendix presents the AttrakDiff method employed in the first empirical study (CHAPTER 3).

	1	2	3	4	5	6	7		
human	○	○	○	○	○	○	○	technical	1
isolating	○	○	○	○	○	○	○	connective	2
pleasant	○	○	○	○	○	○	○	unpleasant	3
inventive	○	○	○	○	○	○	○	conventional	4
simple	○	○	○	○	○	○	○	complicated	5
professional	○	○	○	○	○	○	○	unprofessional	6
ugly	○	○	○	○	○	○	○	attractive	7
practical	○	○	○	○	○	○	○	impractical	8
likeable	○	○	○	○	○	○	○	disagreeable	9
cumbersome	○	○	○	○	○	○	○	straightforward	10
stylish	○	○	○	○	○	○	○	tacky	11
predictable	○	○	○	○	○	○	○	unpredictable	12
cheap	○	○	○	○	○	○	○	premium	13
alienating	○	○	○	○	○	○	○	integrating	14
brings me closer to people	○	○	○	○	○	○	○	separates me from people	15
unpresentable	○	○	○	○	○	○	○	presentable	16
rejecting	○	○	○	○	○	○	○	inviting	17
unimaginative	○	○	○	○	○	○	○	creative	18
good	○	○	○	○	○	○	○	bad	19
confusing	○	○	○	○	○	○	○	cleary structured	20
repelling	○	○	○	○	○	○	○	appealing	21
bold	○	○	○	○	○	○	○	cautious	22
innovative	○	○	○	○	○	○	○	conservative	23
dull	○	○	○	○	○	○	○	captivating	24
undemanding	○	○	○	○	○	○	○	challenging	25
motivating	○	○	○	○	○	○	○	discouraging	26
novel	○	○	○	○	○	○	○	ordinary	27
unruly	○	○	○	○	○	○	○	manageable	28

ANNEX B – USER EXPERIENCE QUESTIONNAIRE

This appendix presents the User Experience Questionnaire (UEQ) employed in the first empirical study (CHAPTER 3).

	1	2	3	4	5	6	7		
annoying	○	○	○	○	○	○	○	enjoyable	1
not understandable	○	○	○	○	○	○	○	understandable	2
creative	○	○	○	○	○	○	○	dull	3
easy to learn	○	○	○	○	○	○	○	difficult to learn	4
valuable	○	○	○	○	○	○	○	inferior	5
boring	○	○	○	○	○	○	○	exciting	6
not interesting	○	○	○	○	○	○	○	interesting	7
unpredictable	○	○	○	○	○	○	○	predictable	8
fast	○	○	○	○	○	○	○	slow	9
inventive	○	○	○	○	○	○	○	conventional	10
obstructive	○	○	○	○	○	○	○	supportive	11
good	○	○	○	○	○	○	○	bad	12
complicated	○	○	○	○	○	○	○	easy	13
unlikable	○	○	○	○	○	○	○	pleasing	14
usual	○	○	○	○	○	○	○	leading edge	15
unpleasant	○	○	○	○	○	○	○	pleasant	16
secure	○	○	○	○	○	○	○	not secure	17
motivating	○	○	○	○	○	○	○	demotivating	18
meets expectations	○	○	○	○	○	○	○	does not meet expectations	19
inefficient	○	○	○	○	○	○	○	efficient	20
clear	○	○	○	○	○	○	○	confusing	21
impractical	○	○	○	○	○	○	○	practical	22
organized	○	○	○	○	○	○	○	cluttered	23
attractive	○	○	○	○	○	○	○	unattractive	24
friendly	○	○	○	○	○	○	○	unfriendly	25
conservative	○	○	○	○	○	○	○	innovative	26

ANNEX C – UX-TIPS

This annex presents the UX-TIPS method we employed in the second empirical study (CHAPTER 3).

Aesthetic Dimension	
Item	**Description**
AST1	The application features a nice and beautiful interface design.
AST2	The color and contrast scheme shown is appropriate.

Emotion Dimension	
Item	**Description**
EMT1	It is pleasant/I like to use the application.
EMT2	The application allows the user to feel happy using it.

Engagement Dimension	
Item	**Description**
EGT1	The application arouses the interest in obtaining it.
EGT2	The application stimulates the desire to recommend it to others.
EGT3	The application stimulates the curiosity to know it more.

Innovative Dimension	
Item	**Description**
INO1	The application has innovative features (different ways to meet the user's need).

Social Dimension	
Item	**Description**

SOC1	The application lets you share information with others.
SOC2	The application allows being always updated (informed) about the contents it provides.
SOC3	The application is known and widely used by other people.

Physical Characteristics Dimension (Applicable for Mobile Applications)		
Item	**Description**	**Do these items apply to the evaluated app?**
PSC1	The application has good battery management (i.e., it does not consume a lot of battery).	( ) Yes ( ) No
PSC2	The application allows/enables the use of sensors to provide interaction in different ways: through GPS (location), accelerometer (movement), gyroscope (gestures) and voice recognition.	( ) Yes ( ) No

Learning and Ease of Use Dimension	
Item	**Description**
LUA1	The application interface is consistent (i.e., same interface items represent the same things).
LUA2	The application content (text, images, information, icons) are displayed in a visible and understandable way.
LUA3	The app's features do what they seem to do.
LUA4	The application is easy enough to perform the activities without difficulties.
LUA5	The application visibly provides tips or guides on how to use it.
LUA6	The application does not require much mental effort to remember how to use it.

Utility Dimension	
Item	**Description**
UTL1	The application assists in an important activity.

Control Dimension	
Item	**Description**
CTR1	The application allows controlling the interaction the way the user wants.

Feedback Dimension	
Item	**Description**
FCK1	The application provides information about the actions the user performs.
FCK2	Information about user actions is objective and understandable.

Dimension Efficiency	
Item	**Description**
EFF1	The application processes the information quickly.
EFF2	The application allows using shortcuts to perform some activities.

Value Added Dimension	
Item	**Description**
VLE1	The application generates value (has benefits that make the user prefer this application over the competitors).

VLE2	The application has/represents values that are important to the user.

Satisfaction Dimension	
Item	**Description**
STF1	The application meets user’s expectations.
STF2	The application fulfills what it is expected to do.

Problem Reporting Form

UX-Tips provides a form to report the problems that users encountered during the UX evaluation. In the first column, users enter the technique item code to which the problem is related. In the second column, the users describe the problems they have encountered.

Table for identified issues

Technique Item Code	Problem Description (Describe the problem you encountered)

ANNEX D – USER EXPERIENCE QUESTIONNAIRE WITH SELF-ASSESSMENT MANIKIN

This annex presents the User Experience Questionnaire (UEQ) added with the valence dimension from Self-Assessment Manikin (SAM) employed in the second empirical study (CHAPTER 3).

In this second empirical study, we employed the shortened version of UEQ (Schrepp et al., 2017), added with the valence dimension from SAM (Self-Assessment Manikin) (Bradley and Lang, 1994) for getting participants' overall satisfaction with the application they used. The questionnaire comprised two parts. The first part (below) was filled only by the participants who had used similar applications before. The second part (next page) was filled by all participants.

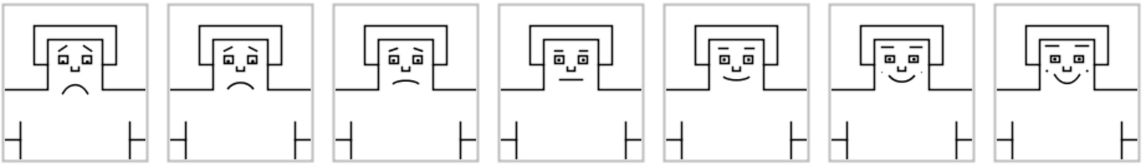
Name:____

Do you use any application similar to the app you just used? Which one? __________________

Please rate your experience with the **APP YOU WROTE ABOVE:**

clear	○	○	○	○	○	○	○	confusing	1
inneficient	○	○	○	○	○	○	○	efficient	2
complicated	○	○	○	○	○	○	○	easy	3
obstructive	○	○	○	○	○	○	○	supportive	4
boring	○	○	○	○	○	○	○	exciting	5
not interesting	○	○	○	○	○	○	○	interesting	6
conventional	○	○	○	○	○	○	○	inventive	7
usual	○	○	○	○	○	○	○	leading edge	8

OVERALL SATISFACTION WITH THE APP YOU WROTE ABOVE:

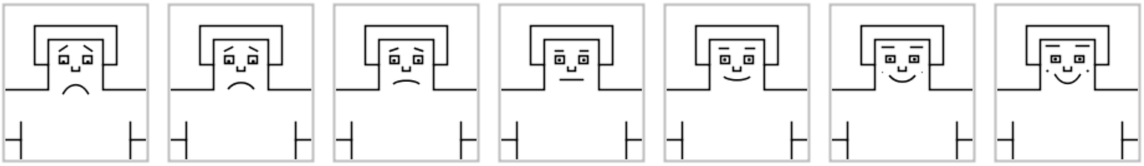


○ ○ ○ ○ ○ ○ ○

Now evaluate the experience you had using **THE APP FROM THIS STUDY:**

clear	○	○	○	○	○	○	○	confusing	1
inneficient	○	○	○	○	○	○	○	efficient	2
complicated	○	○	○	○	○	○	○	easy	3
obstructive	○	○	○	○	○	○	○	supportive	4
boring	○	○	○	○	○	○	○	exciting	5
not interesting	○	○	○	○	○	○	○	interesting	6
conventional	○	○	○	○	○	○	○	inventive	7
usual	○	○	○	○	○	○	○	leading edge	8

OVERALL SATISFACTION WITH THE APP FROM THIS STUDY:



○ ○ ○ ○ ○ ○ ○