ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

THESIS PRESENTED TO
ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
Ph.D.

BY
Fatih NAYEBI

iOS APPLICATION USER RATING PREDICTION USING USABILITY EVALUATION AND MACHINE LEARNING

MONTREAL, "JUNE 10, 2015"

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS:

Mr. Alain Abran, thesis director
Department of Software Engineering and Information Technologies

Mr. Jean-Marc Desharnais, co-advisor
Department of Software Engineering and Information Technologies

Mr. Michel Rioux, committee president
Department of Automated Manufacturing Engineering

Mme. Cherifa Mansoura Liamani, external examiner
Senior consultant at National Bank of Canada

Mr. Witold Suryn, invited examiner
Department of Software Engineering and Information Technologies

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND THE PUBLIC

ON "JUNE 8, 2015"

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

ACKNOWLEDGEMENTS

I dedicate this study to my father who passed away 14 years ago; he always conducted me into critical thinking, research, science, and studying. I would like to thank my family and especially my mother for their endless support and kindness: I would not have accomplished any of my goals if they had not been with me all through my study, and throughout my life.

I am very thankful to my thesis director Dr. Alain Abran. He has inspired me with his intelligence and scientific insight. Assisted with his invaluable comments and guidelines, I was able to conclude this study successfully. He will be the one to follow as my role model in my future academic life.

I am grateful to my co-advisor Dr. Jean-Marc Desharnais, who helped me as a friend and provided support through all of my Ph.D. studies and my academic life. He has participated and assisted in many of the experiments for this research project: this has been of a great help.

I am also greatly thankful to my wife for her support and friendship throughout my Ph.D. studies.

I also want to thank the members of GÉLOG Software Engineering Research Laboratory at École de technologie supérieure, and in particular Gustavo Adolfo Vasquez, for their valuable support and friendship.

PRÉDIRE L'ÉVALUATION DES APPLICATIONS iOS PAR LES UTILISATEURS AVEC LES TECHNIQUES D'ÉVALUATION DE L'UTILISABILITÉ ET L'APPRENTISSAGE AUTOMATIQUE

Fatih NAYEBI

SUMMARY

Les applications mobiles gagnent en popularité en raison des avantages significatifs des appareils mobiles, tels que la portabilité, la capacité de détecter leur emplacement, l'identité électronique et l'appareil photo intégré. Cependant, ces dispositifs ont un certain nombre d'inconvénients en termes de facilité d'utilisation, comme la limitation des ressources (ex. mémoire) et la taille de l'écran.

Un certain nombre d'études ont étudié les défis de la facilité d'utilisation dans un contexte des appareils mobiles et des définitions ont été proposées pour la mesure de l'utilisabilité des applications mobiles. L'évaluation de l'utilisabilité des applications pour les appareils mobiles est une étape cruciale pour traiter ces difficultés et pour réussir sur les marchés de l'application mobile, aux l'App Store d'Apple. L'évaluation de la facilité d'utilisation doit être adaptée à tous les divers systèmes d'exploitation des appareils mobiles utilisés, car ils ont chacun leurs propres caractéristiques particulières.

Le 'App Store' d'Apple est la seule source pour l'achat et l'installation des applications iOS. Les utilisateurs évaluent les applications dans l'App Store et tiennent compte des évaluations des autres utilisateurs quand ils décident d'acheter une application. Notre hypothèse est que les utilisateurs ont tendance à donner des cotes plus élevées pour les applications qui satisfont à leurs exigences fonctionnelles et non-fonctionnelles. Une autre hypothèse est que la facilité d'utilisation est une des exigences non fonctionnelles que les utilisateurs considèrent quand ils cotent les applications. Par conséquent, il est important de développer des applications dont la facilité d'utilisation est plus élevée et d'améliorer l'expérience utilisateur pour réussir dans l'App Store. Apple publie une ligne directrice nommée "Human Interface Guidelines (HIG)" et recommande de suivre ces directives lors de la conception des applications iOS et leur développement. Il y a également d'autres lignes directrices dans la littérature qui suggèrent différents principes de conception ainsi que des heuristiques. Malheureusement, la relation entre ces lignes directrices et le succès de l'App Store est inconnue.

Enfin, il n'existe pas une méthode formelle pour prédire la réussite d'une application dans l'App Store. Les développeurs ou sociétés de développement passent beaucoup de temps pour développer une application sans avoir la moindre idée de la réussite de leur application dans l'App Store.

Ce projet de recherche combine les lignes directrices de la littérature et propose un modèle d'évaluation des applications iOS pour évaluer la facilité d'utilisation des applications iOS.

Il présente ensuite une analyse de la relation entre les critères et la cote des utilisateurs de l'application dans l'App Store pour 99 application iOS. Ce projet de recherche propose également un modèle d'apprentissage automatique pour prédire la réussite d'une application iOS dans l'App Store, le modèle étant basé sur la méthode d'évaluation proposée dans la première partie de cette recherche.

Mots clés: Interaction homme-machine, utilisabilité, mesure des logiciels, génie logiciel, apprentissage automatique et de développement des applications mobiles

iOS APPLICATION USER RATING PREDICTION USING USABILITY EVALUATION AND MACHINE LEARNING

Fatih NAYEBI

ABSTRACT

Mobile applications are earning popularity because of the significant benefits of smartphones, such as: portability, location awareness, electronic identity, and an integrated camera. Nevertheless, these devices have a number of disadvantages concerning usability, such as limited resources and small screen size.

A number of studies have investigated usability challenges in a mobile context and proposed definitions and measurement of the usability of mobile applications. Evaluating the usability of applications for mobile operating systems is a crucial step in addressing these difficulties and achieving success in mobile application markets, such as Apple's App Store. Usability evaluation must be tailored to the various mobile operating systems in use, each with its particular characteristics.

Apple App Store is the only source for buying or installing iOS applications. Users rate applications in the App Store and take into the account other users' evaluations when buying an application. Users tend to give higher ratings to the applications that satisfy their functional and non-functional requirements. Usability is one of the non-functional requirements that users consider when they rate applications. Hence, it is important to develop applications with higher usability and better user experience to be successful in the App Store.

Apple publishes a guideline named "HIG (Human Interface Guidelines)" and recommends following these guidelines during the design and development of iOS applications. There are also other guidelines that suggest other design principles and heuristics but the relationship between these guidelines and App Store success is unknown.

In addition, there is not any explicit method to predict the success of an application in the App Store. Developers and development companies spend much time to develop an application with little clue about the success of their application in the App Store.

This research project combines the guidelines from the literature and proposes an iOS application usability evaluation model to evaluate iOS application usability. It presents next an analysis of the relationship between criteria and application's App Store user rating by evaluating 99 applications. This research project also proposes a machine learning model to predict the success of an iOS application in the App Store, based on the evaluation method proposed in the first part of this research.

Keywords: Software Engineering, Machine Learning, Mobile Human Computer Interaction, Usability Evaluation, and Mobile Application Development

CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF ABREVIATIONS

ETS	École de Technologie Supérieure
ISO	International Standard Organization
HIG	Apple Human Interface Guidelines
APP	iOS Application
UI	User Interface
WAVG	Weighted Average
TP	True Positives
TN	True Negatives
FP	False Positives
FN	False Negatives
SVM	Support Vector Machines
MLP	Multi-Layer Perceptron
RBF	Radial Basis Function

INTRODUCTION

Smartphones and applications provide significant benefits to their users in terms of portability, location awareness, and accessibility. Smartphones' prices decreases and improvements in hardware and software capabilities of smartphones have led to the expansion of smartphones and related markets.

Sophisticated mobile applications find their way into everyday life with a much broader customer base. This has made usability more critical than before because of varying user types and their usability requirements. Therefore, developers and development companies perceive the criticality of designing and developing their applications with user-oriented approaches instead of technology-oriented approaches. Nowadays, developers and development companies examine, for better understanding, the interactions between a user and the application.

Individual developers and development companies can quickly distribute their applications via application markets such as Apple App Store (Apple, 2015). Due to the popularity of smartphones and the ease of application distribution, the development of applications for these platforms has gained much momentum and market share.

Nowadays Apple App Store has more than 1.3 million applications (Statistia, 2015) aiming at various usage scenarios in different categories. Users can explore different categories to examine their desired applications. Having many choices makes finding appropriate applications more time-consuming and harder for the user. Users tend to read other users' reviews and examine application ratings to find the appropriate applications.

On the other hand, the increased number of mobile applications in the market challenges developers to be successful in the competition for developing higher quality applications and taking into account what users like, and why they give better ratings.

Understanding why users give high or low ratings in the App Store to specific applications is not a trivial task. Users may give better ratings to mobile applications that are easy to learn, take less time to complete a particular task and look friendlier because users are less computer-

oriented nowadays. Although, different users may have different motivations and expectations, one expects that users give higher ratings to the applications that satisfy their requirements.

Functional and non-functional requirements are two generic types of software requirements. Requirement analysis specialists specify necessary functional requirements of applications by eliciting the requirements from their users and analyzing same category applications in the application market. Requirement analysis specialists work with user experience designers to specify non-functional requirements. User experience designers design applications regarding specific guidelines. Quality assurance professionals and application usability examiners evaluate particular non-functional requirements of the application. For instance, application usability examiners can evaluate an application in terms of quality in use, usability, performance and esthetics.

Usability is one of the essential aspects of quality and directly affects the user satisfaction (ISO/IEC25010, 2011), consequently the user given ratings. Researchers have investigated the usability aspects to be evaluated and measured (Nielsen, 1994; Sauro and R.Lewis, 2012; Tullis and Albert, 2013). In addition to these studies, usability measurement and analysis methods have been proposed. Laboratory experiments, field studies, and expert-based evaluations are some of the most significant strategies applied by researchers (Ryu and Smith-Jackson, 2006).

However, each evaluation method has its benefits and drawbacks. Some of the usability evaluation methods are difficult to apply, and some others are dependent on the evaluator's opinions or instruments. In addition to these challenges and needs, smartphones and mobile applications have been changing very fast. Also, smartphones and applications' usability differ from other computer systems such as desktop and web because of the various characteristics and aspects of mobile devices. Some of mobile applications are embedded into the smartphones during manufacturing and some are installed by customers from mobile software distribution platforms such as Apple App Store (Apple, 2015) and Google Play android market.

The architecture of a mobile application must take into account the operating system, mobile application distribution market, user experience design of embedded applications and a number of design constraints such as limited resources, connectivity, data entry UI controls, and different display resolutions and sizes of mobile devices.

The literature proposes various guidelines and heuristics to designers and developers. Apple provides HIG (Human Interface Guidelines) particularly for APPs (iOS applications) that cover platform characteristics, human interface principles, and user experience guidelines (Apple, 2013).

Designers and developers rely on the guidelines and heuristics to deliver a higher quality application when releasing an application to the market and expecting to achieve high user ratings. Due to the quantity and diversity of these guidelines, professionals with mobile application development and user experience knowledge in other words application usability examiners are required to examine all the aspects of application design in a usability evaluation process before submitting an application to the App Store.

Application usability examiners need to select a proper combination of suggested guidelines and iOS HIG and apply them during the APP design. The aforementioned is not a trivial task owing to the variety of guidelines and possible contradictions between them. APP usability experts can use these guidelines to evaluate applications as well, but they may require to formulate them as evaluation criteria.

The relationship between these guidelines and the App Store success is not clear. Furthermore, it is not proven if evaluating applications against criteria formulated from guidelines have a significant correlation with user ratings in the App Store. Finding the relationship would be very beneficial for mobile application development companies therefore they could direct their efforts on the specific guidelines and evaluation criteria that are most relevant.

Even if knowing the most appropriate guidelines and applying them to application design will help the designers and developers, it is not enough to foresee if an application will have high ratings in the market.

Predicting if the application will have positive user rating or negative user rating before releasing it to the App Store could be helpful for individual developers and development companies. In the case that predicted user ratings are negative, they can revisit their design and improve it further because the first version of an application is the most crucial version in the App Store. Having bad reviews and low user ratings for the first version of an application will make a substantial unfavorable impression about the application. Solving the issues in the future releases may not be enough to increase user ratings and restore the customer trust.

Predicting the user rating of an application in the App Store before releasing it to the App Store can be a two-class classification problem (Alpaydin, 2014): in other words, Apple App Store iOS applications can be classified as applications that have Positive user rating (TRUE class label) or Negative user rating (FALSE class label).

This research study aims to provide a model and related artifacts for the user rating prediction of Apple App Store iOS applications. The structure of this document is as follows:

- Chapter 1 summarizes the literature on the evaluation of mobile application usability.
- Chapter 2 summarizes the literature on some of the machine learning classification models for the user rating prediction.
- Chapter 3 presents the problem statement, research motivation and objectives of the thesis.
- Chapter 4 presents the research methodology of the thesis including the user rating prediction model for the Apple App Store iOS applications.
- Chapter 5 presents the required artifact for the usability evaluation of mobile applications including:
    - The definition of usability in the literature and ISO standards

  - The guidelines to consider during application design including usability heuristics
  - The mobile human interface guidelines and Apple HIG
  - The usability evaluation methods including expert-based evaluation and user-based evaluation
  - The common usability study scenarios
  - The standardized usability questionnaires
  - The usability measures defined in ISO standards and the literature.
- Chapter 6 presents the model proposed for the usability evaluation of iOS applications and covers the following aspects:
  - Expert-based iOS application usability evaluation criteria
  - User-based iOS application usability evaluation criteria
  - Apple App Store user rating evaluation.
- Chapter 7 presents an iOS application usability evaluation study including the following artifacts:
  - Evaluation process
  - Dataset and data analysis results
  - Summary of the dataset analysis.
- Chapter 8 presents the experiments on the user rating prediction of the Apple App Store iOS applications including the following:
  - Experiments with five different models
  - Prediction model comparison
  - Best model selection
  - Threats to validity.
- Chapter 9 summarizes the key contributions and presents the future work.

- The Conclusion chapter concludes the study.

In addition to the above aforementioned chapters, appendices present the following artifacts:

- Appendix I: The State of the Art of Mobile Application Usability Evaluation - A publication by the researchers of this study.
- Appendix II: An Expert-based Framework for Evaluating iOS Application Usability - A publication by the researchers of this study.
- Appendix III: List of publications selected and reviewed in the systematic literature review of this study.
- Appendix IV: List of the 99 selected iOS application for the experimentation.
- Appendix V: ISO 9241 - Ergonomic Requirements for Office Work with Visual Display Terminals.
- Appendix VI: Guidelines to consider during application design except the usability heuristics.
- Appendix VII: Apple HIG user experience guidelines.
- Appendix VIII: Standardized usability questionnaires for user-based usability evaluation.
- Appendix IX: Common usability study scenarios to be used in user-based application usability evaluations.
- Appendix X: List of cited-by publications.

CHAPTER 1

LITERATURE REVIEW ON iOS APPLICATION USABILITY EVALUATION

This chapter presents a review of the literature for mobile application usability evaluation and analyzes the results and findings regarding iOS application usability.

Several studies have reported evaluations and comparisons of usability evaluation methods. Coursaris and Kim (2006) made a meta-analytical review of empirical mobile usability studies and presented a usability evaluation framework adapted to the context of a mobile computing environment. Using this framework, Coursaris (2011) conducted a qualitative meta-analytical review of more than 100 empirical mobile usability studies.

Most of the reviewed studies are informal literature reviews or comparisons without defined research questions, search process, defined data extraction or data analysis process. In addition, the majority of studies examine generic usability evaluation methods; only a few studies are particularly concentrated on mobile application usability evaluation methods (Coursaris, 2011).

1.1 Need for a systematic review on iOS application usability evaluation

Although there are several studies concerning usability evaluation methods, no systematic review study has been published in the field of mobile application usability apart from Coursaris (2011). Also Coursaris (2011) did not look at the recent changes in mobile applications. Also, it did not review some of our review questions. Therefore, the systematic review presented in this chapter considers newly published articles with specific questions.

1.2 Systematic review methodology

This section reports on our systematic review conducted considering the guidelines provided in the literature such as Budgen *et al.* (2008), Petersen *et al.* (2008), and Kitchenham (2007). A systematic review study is a means of categorizing and summarizing the existing information about a research question in an unbiased manner. Our systematic review study was

performed in three stages: Planning, Conducting, and Reporting. The activities concerning the planning and conducting stages of our systematic review study are described in the following sub-sections, and the results are presented in section 1.3.

1.2.1 Planning stage

In this stage, the following activities have been performed in order to establish a review protocol:

- Establishment of the review question
- Definition of the search strategy
- Selection of primary studies
- Definition of the data extraction strategy
- Selection of synthesis methods

The following sub-sections describe these activities in detail.

1.2.1.1 Review questions

The goal of this systematic review is to examine the current use of usability evaluation methods in mobile application development regarding the following research questions:

- Q1 - Which usability heuristics were considered by researchers during mobile application usability evaluation?
- Q2 - Which usability evaluation methods have been employed by researchers to evaluate mobile applications?
- Q3 - Which types of mobile usability studies have been conducted by researchers?

- Q4 - Which types of usability measures have been employed by researchers to evaluate mobile applications?
- Q5 - In which phases and which mobile artifacts usability evaluation methods are applied?
- Q6 - Is a feedback provided by the usability evaluation methods?
- Q7 - Which empirical validation of the usability evaluation methods are applied?
- Q8 - Does the study consider the APP (Apple iOS application) specific characteristics?
- Q9 - Does the study consider the Apple App Store user ratings?

The review questions will empower categorizing and summarizing the current knowledge about mobile application usability evaluation, distinguishing gaps in current research and will propose fields for further examination and will give useful knowledge to beginner usability practitioners. Because the proposed review questions are too general, they have been decomposed into more detailed sub-questions to be addressed.

1.2.1.2 Search strategy

This systematic study has used the following digital libraries to search for the primary studies:

- IEEEXplore
- ACM Digital Library
- Springer Link
- Science Direct

To perform an automatic search on the selected digital libraries, a search string (see Table 1.1) consisting of four parts to cover the concepts representing the mobile application usability evaluation is utilized:

a. The first part is related to the mobile field studies

b. The second part is related to the application area studies

c. The third part is similar to studies that present usability

d. The fourth part is related to evaluation methods

Table 1.1 presents the search string: the Boolean OR has been utilized to join other terms and equivalents in each main section, and Boolean AND has been employed to join the main sections. The search was conveyed by using the search string to the title, abstract and keywords of each article for all the sources.

The reviewed period included studies published from 1990 to April 2012. As the search was performed in April 2012, publications pertaining to April 2012 and later are not considered in this systematic review study.

Table 1.1 Literature Review search queries and result counts

No	Search	Results
1	mobil* app* usabilit* eval*	754
2	mobil* usabilit* evaluation	828
3	mobil* usabilit* measure*	336
4	mobil* app* usabilit* measure*	209

1.2.1.2.1 Search details

Search Notation: Truncation (*) is used to search for words that begin with the same letters. evalu* returns evaluation, evaluate, evaluating...

- Results: 837 (565 Compendex, 272 Inspec)
- Search sources: engineeringvillage2.org
- Search date: 20th April 2012

- Search language: English

The applied search string consists of the following artifacts:

- (mobile OR mobility) AND
- (app OR application) AND
- (usability OR usable) AND
- (evalu* OR assess* OR measur* OR experiment* OR stud* OR test* OR method* OR techni* OR approach*)

1.2.1.2.2 Searching sources

The systematic study has used the Compendex and Inspec databases.

Compendex: Compendex is an extensive bibliographic database of scientific and technical engineering research available, comprising all engineering disciplines. It covers millions of bibliographic citations and abstracts from various engineering journals and conference proceedings. When combined with the Engineering Index Backfile (1884-1969), Compendex covers well over 120 years of core engineering literature.

Inspec: Inspec combines bibliographic citations and indexed abstracts from publications in various fields, including computer science, information technology, and software engineering.

1.2.1.3 Selection of primary studies

This systematic study considers the title, abstract and keywords for all automated search retrieved studies to decide whether or not to cover in the review and includes studies meeting at least one of the following inclusion criteria:

- Study presents the definition of mobile application usability evaluation methods.

- Study reports mobile application usability evaluations through the employment of existing usability evaluation methods.

This study excludes the studies meeting at least one of the following exclusion criteria:

- Study does not focus on the mobile application field.
- Study presents only recommendations, guidelines, or principles for mobile application design.
- Study presents only usability attributes and their associated measures.
- Study presents only accessibility studies for disabled users.
- Study presents methods on how to summarize usability measures.
- Study presents functional evaluation processes.
- Study is an introduction to particular issues, and workshops.
- Study is a duplicate report of the same research in other sources.
- Study is not written in English.

Most relevant references of the selected studies were followed to find and include other pertinent studies.

1.2.1.4 Data extraction strategy

This study employs a data extraction strategy based on the set of possible answers for each defined research sub-question. The reasonable answers for each research sub-question are explained in more detail as follows.

Considering the Q1 (Usability heuristics employed by researchers), a study can be classified in one or more of the following answers (Nielsen, 1994; Gerhardt-Powals, 1996; Shneiderman *et al.*, 2010; Weinschenk and Barker, 2000):

a. Simple and Natural Dialogue

b. Speak the users' language

c. Minimizing user memory load

d. Consistency

e. Feedback

f. Clearly marked exits

g. Shortcuts

h. Good error messages

i. Prevent errors

j. Help and documentation

Considering the Q2 (Types of usability evaluation method employed), a study can be classified in one or more of the following answers (Tullis and Albert, 2013; Sauro and R.Lewis, 2012):

a. Heuristic evaluation

b. Performance measures

c. Thinking aloud

d. Observations

e. Questionnaires and Interviews

f. Focus groups

g. Logging actual use

h. User feedback

Considering the Q3 (Types of Mobile usability studies applied), a study can be classified in one or more of the following answers (Tullis and Albert, 2013; Sauro and R.Lewis, 2012):

a. Completing a transaction
b. Comparing applications
c. Evaluating frequent use of the same application
d. Evaluating navigation and/or information architecture
e. Increasing awareness
f. Problem discovery
g. Maximizing usability for a critical application
h. Creating an overall positive user experience
i. Comparing alternative designs

Considering the Q4 (Types of usability measures employed), a study can be classified in one or more of the following answers(Tullis and Albert, 2013; Sauro and R.Lewis, 2012):

a. Task success
b. Task time
c. Errors
d. Efficiency
e. Learnability
f. Issues-based measures
g. Self-reported measures

h. Behavioral and physiological measures

Considering the Q5 (Phase(s) and mobile artifacts in which the usability evaluation methods are applied), a study can be classified into one or more ISO/IEC12207 (2008) high-level processes:

a. Requirements: if the artifacts that are used as input for the evaluation include high-level specifications of the mobile application (e.g., task models, uses cases, usage scenarios).

b. Design: if the evaluation is conducted on the intermediate artifacts that are created during the mobile application development process (e.g., navigational models, abstract UI models, dialog models).

c. Implementation: if the evaluation is conducted at the final UI or once the mobile application development is completed.

Considering the Q6 (Feedback provided by the usability evaluation methods), a study can be classified into one of the following answers:

a. Yes: if the usability evaluation method provides recommendations or guidance to the designer on how to correct the detected usability problems.

b. No: if the usability evaluation method aims only at reporting usability problems.

Considering the Q7 (Empirical Validation of the usability evaluation methods) one can categorize a study to one of the following categories, considering the objective of the validation and the circumstances of empirical investigation (Fenton and Pfleeger, 1998):

a. Survey: if a study provides a review of previous researches that has been in use for a period to gather feedback about the advantages and shortcomings of the usability evaluation method.

b. Case study: if a study provides an observation in which data is collected to evaluate the performance of the usability evaluation methods.

c. Controlled experiment: if a study provides a formal, accurate, and controlled examination based on verifying hypotheses concerning the performance of the usability evaluation methods.

d. No: if it does not provide any validation or if it only presents a proof of concept.

Considering the Q8 (Consideration of the APP specific characteristics), a study can be classified into one of the following answers:

a. Yes: if the evaluation is conducted on the APP (iOS application) specific characteristics such as multi-touch gestures, device orientation changes, high-resolution screen needs and location-awareness.

b. No: if the study does not provide any consideration regarding APP (iOS application) specific characteristics.

Considering the Q9 (Consideration of Apple App Store user ratings), a study can be classified into one of the following answers:

a. Yes: if the study considers the Apple App Store user ratings and employs the user ratings in the research.

b. No: if the study does not consider the Apple App Store user ratings.

Table 1.2 presents the questions and possible answers.

Table 1.2 Literature Review questions and possible answers

Questions	**Possible Answers**
Q1: Usability heuristics employed by researchers	(a) Simple and Natural Dialogue (b) Speak the users' language (c) Minimizing user memory load (d) Consistency (e) Feedback (f) Clearly marked exits (g) Shortcuts (h) Good error messages (i) Prevent errors (j) Help and documentation
Q2: Types of usability evaluation method employed	(a) Heuristic evaluation (b) Performance measures (c) Thinking aloud (d) Observations (e) Questionnaires and Interviews (f) Focus groups (g) Logging actual use (h) User feedback
Q3: Types of Mobile usability studies applied	(a) Completing a transaction (b) Comparing applications (c) Evaluating frequent use of the same application (d) Evaluating navigation and/or information architecture, Increasing awareness (f) Problem discovery (g) Maximizing usability for a critical application (h) Creating an overall positive user experience (i) Comparing alternative designs
Q4: Types of usability measures employed	(a) Task success (b) Task time (c) Errors (d) Efficiency (e) Learnability (f) Issues-based measures (g) Self-reported measures (h) Behavioral and physiological measures
Q5: Phases and mobile artifacts in which the usability evaluation methods are applied	(a) Requirements (b) Design (c) Development
Q6: Feedback provided by the usability evaluation methods	(a) Yes (b) No
Q7: Empirical validation of the usability evaluation methods	(a) Survey (b) Case Study (c) Experiment (d) No
Q8: Consideration of the APP specific characteristics	(a) Yes (b) No
Q9: Consideration of Apple App Store user ratings	(a) Yes (b) No

1.2.2 Conducting stage

The application of the review protocol yielded 296 results. Next, 132 research papers were selected after examining the yielded results in accordance with the inclusion criteria. The followings explain the several issues found at this stage:

- Some studies had been published in more than one journal/conference. Therefore a complete version of the study has been selected.
- Some studies had appeared in more than one source. Therefore, they were taken into account only once according to the search order: IEEEXplore, ACM, Springer Link and Science Direct.
- The search results revealed that the mobile usability research studies had been published in various conferences/journals related to different fields such as Mobile Application Development, Software Engineering, Human-Computer Interaction (HCI), and other related fields.

1.3 Results

132 studies were reviewed for more details to pertain the relevant studies about Mobile Application Usability Evaluation. Hence, 38 studies qualified as final studies to take part in this review study. Appendix III lists all the qualified studies.

1.4 Principal findings

This section presents the observations on the 38 identified studies that use either a user-based field study or an expert-based evaluation methodology. Three of the primary studies (Ryu, 2005; Gafni, 2009; Hussain and Kutar, 2009) propose measures and measurement methods for mobile usability evaluation (Nayebi *et al.*, 2012). The following sub-sections present the principal findings on the the 38 identified studies.

1.4.1 Limitations of the lab experiments

All methodologies have benefits and drawbacks, and one of the most vital differentiative criteria is the cost of experimentation. Laboratory experiments require particular instruments and are consequently more costly than other methodologies. Duh and Tan (2006) compared laboratory and field experimentation methodologies and claimed the following: "There were many more types and occurrences of usability problems found in the field than in the laboratory. Those problems discovered tend to be critical issues regarding usability. Some of these usability problems are only related to the device being used in the field, which could not be found using conventional laboratory usability tests. With regards to the users' behaviors, users behave less positively and more negatively in the field than in the laboratory. Some behaviors can only be observed in the field. Users also take longer time to perform certain tasks and also present more negative feelings, such as dissatisfaction and difficult of use, to the use of the device in the field."

1.4.2 New mobile operating system needs

This study observes that the questionnaires and expert-based methods developed for mobile usability evaluation do not investigate the features provided in the newest mobile operating systems such as Apple iOS. For instance, none of the reviewed studies examine the multi-touch gestures (e.g. the tap, flick, pinch, and slide), device orientation changes, high-resolution screen needs (e.g. retina displays) and location awareness capabilities. In fact, there is not any particular study about usability evaluation of APPs (iOS applications).

1.4.3 Usability measures

The literature and ISO defined measures discussed in the identified studies are often not accurately defined. It is not clear in Ryu (2005) on what grounds users answer questions from questionnaires, such as, "Does the product have all the functions and capabilities you expect it

to have?" This question is conceptual, and so the answer will be subjective and highly dependent on the user's judgment.

1.4.4 Measurement methods

The measurement methods as described in Abran (2010) are not accurately explained for the collected data. For instance, in Hussain and Kutar (2009) it is not obvious whichever measurement methods were utilized to gather data such as: "Number of system resources displayed," "Number of voice assistance in a task".

1.4.5 Single evaluation methodology

Published studies rely on a single methodology. However, a user-based, along with an expert-based evaluation, could be more informative in mobile usability evaluation studies. In this case, the user-based field study is based on the experience of users while a measurement/evaluation expert performs the expert-based evaluation. Combining the two methodologies should provide more significant information for evaluation.

1.4.6 iOS application usability evaluation criteria

There is not any set of criteria and corresponding evaluation methods covering popular mobile operating systems such as Apple iOS, which can be utilized for benchmarking purposes specifically to compare the usability of APPs (iOS applications).

1.4.7 Apple App Store user ratings

User ratings in mobile application markets, such as Apple App Store and Google's Android Play Store, are helpful resources for usability studies; however, the considered studies did not study mobile application market ratings. In fact, there is not any study considering Apple App Store user ratings.

1.5 Conclusions of the literature review

In summary, this systematic study has examined the mobile application usability evaluation criteria and methods used in user-based and expert-based evaluations and discovered the limitations of the lab experiments.

In addition to this, the systematic study has discovered that there is not any study approaching the requirements of new mobile operating systems such as multi-touch gestures, device orientation changes, high-resolution screen needs and location-awareness capabilities. Indeed, there is not any particular APP (iOS application) usability evaluation study.

Furthermore, this systematic study has revealed that none of the studies consider and employ the Apple App Store user ratings.

From the findings of this systematic literature review, the following research needs were identified:

A) Expert-based APP usability evaluation model:

A large number of applications have been developed for Apple iOS and published in the Apple App Store. Apple provides HIG (Human Interface Guidelines) specifically for APPs design. Apple HIG consists of iOS specific guidelines that can be used to construct the expert-based APP usability evaluation criteria. APP usability experts may use APP usability evaluation model to evaluate and identify usability issues of the APP before submitting it to the App Store.

B) User-based APP usability evaluation model:

A user-based APP usability evaluation mode is required that regards the form of a questionnaire that can reflect new mobile operating system requirements, such as interaction with a multi-touch screen, displays of different resolutions and dimensions, device orientation changes, and gestures corresponding tap, flick, slide, and pinch.

C) Apple App Store user rating prediction model:

There is already plenty of data on mobile application user ratings and satisfaction on the mobile application markets such as Apple App Store that can be employed for the analysis and prediction model construction.

CHAPTER 2

LITERATURE REVIEW ON MACHINE LEARNING CLASSIFICATION MODELS

Classification is the problem of identifying to which class (In this study, TRUE: Positive user rating and FALSE: Negative user rating) a new instance (In this study, an iOS application) belongs.

There are many different machine learning models to solve the classification problem (Witten *et al.*, 2011; Han *et al.*, 2011; Alpaydin, 2014). Reviewing all of the models is out of scope of this research. Also, there are publications regarding different machine learning classification models. For instance, Fernández-Delgado *et al.* (2014) evaluates 179 different classification models from varying following families:

a. Discriminant Analysis

b. Bayesian Classification

c. Neural Networks

d. Support Vector Machines

e. Decision Trees

f. Rule-based Classifiers

g. Boosting

h. Bagging

i. Stacking

j. Random Forests

k. General Linear Models

l. Nearest Neighbors

m. Partial Least Squares and Principal Component Regression

n. Logistic and Multinomial Regression

o. Multiple Adaptive Regression Splines

Any machine learning model belong to the above families could be a model candidate in this study. However, this study employs only five of these models belonging to three families as follows:

a. Bayesian

- Naïve Bayes
- Bayesian Network

b. Neural Networks

- MLP (Multi-Layer Perceptron)
- RBF (Radial Basis Function)

c. SVM (Support Vector Machines)

Therefore, the following sections present the related models.

2.1 Bayesian Classification

Bayesian Statistics produces a framework for building intelligent learning systems to solve problems such as classification. Bayesian machine learning models can predict class membership probabilities such as the probability that a given set of criteria belongs to a particular class (In this study, TRUE class label: Positive user rating and FALSE class label: Negative user rating).

Bayesian classification is based on Bayes' theorem.

Bayes theorem states that:

$$P(M|D) = \frac{P(D|M)P(M)}{P(D)} \tag{2.1}$$

Also,

$$P(D,M) = P(M)P(M|D) = P(M)P(D|M) \tag{2.2}$$

where $P(D)$ is the probability of D, $P(D|M)$ is the conditional probability of D given M and $P(D,M)$ is the joint probability of D and M

It is possible to read the equation 2.1 in the following manner: "the probability of the model given the data $P(M|D)$ is the probability of the data given the model $(P(D|M))$ times the prior probability of the model $P(M)$ divided by the probability of the data $(P(D))$".

Bayesian Statistics, particularly, the Cox's theorem (Cox, 1946) implies that Bayes rule can be used to describe and manipulate the degree of belief in models or hypotheses. In other words, degrees of beliefs should be treated in exactly the same way as probabilities. Hence, the prior $P(M)$ above represents numerically the degree one believes model M to be the actual model of the data before actually observing the data, and the posterior $P(M|D)$ represents the extent one believes model M after observing the data.

2.1.1 Bayes rule applied to Machine Learning

Machine learning can be described as learning models of data. The Bayesian framework for machine learning states that one can start by enumerating all reasonable models of the data and assigning the prior belief $P(M)$ to each of these models. Then, upon observing the data D, one can evaluate how likely the data was under each of these models to compute $P(D|M)$ (Likelihood of M). Multiplying this likelihood by the prior $(P(M))$ and renormalizing results in the posterior probability over models $P(M|D)$ which encloses everything that one has learned from the data respecting the possible models under consideration.

In practice, applying Bayes rule is not very practical because it involves summing or integrating over a very large space of models. These computationally intensive sums or integrals can be avoided by using approximate Bayesian methods. Laplace's approximation (Azevedo-Filho and Shachter, 1994), variational approximations, expectation propagation, and Markov Chain Monte Carlo methods are some examples of approximate Bayesian methods (Bishop, 2006).

2.1.2 Bayesian approach to Machine Learning

In the Bayesian approach to machine learning the following steps take place:

2.1.2.1 Formulating the knowledge about the situation probabilistically

In Bayesian Machine Learning model building, it is required to define a model that expresses qualitative aspects of knowledge by forms of distributions and independence assumptions. The model will have some parameters that are unknown. A prior probability distribution needed to be specified to express the beliefs about likely values of the unknown parameters.

2.1.2.2 Dataset construction

A dataset is essential for Bayesian Machine Learning: therefore a problem related dataset is required to be gathered and provided to the model.

2.1.2.3 Computing the posterior probability distribution for the criteria, given the dataset

The posterior distribution for the model parameters given the dataset can be found by combining the prior distribution with the likelihood for the parameter given the data.

This is done using Bayes' rule:

$$P(parameters|data) = \frac{P(parameters)P(data|parameters)}{P(data)} \tag{2.3}$$

The denominator is just the required normalizing constant and can often be filled in at the end if necessary. So as a proportionality, it can be written as:

$$P(parameters|data) \propto P(parameters)P(data|parameters) \tag{2.4}$$

which can be written schematically as:

$$Posterior \propto Prior \times Likelihood \tag{2.5}$$

2.1.2.4 Representing the prior and posterior distributions by samples

The complex distributions that are often used as priors, or obtained as posteriors, may not be easily expressed or understood using formulas. A standard technique is to represent a distribution by a sample of many values drawn randomly from it. It is possible to visualize the distribution by viewing these sample values, or low-dimensional projections of them. In addition, it is possible to make Monte Carlo (Bishop, 2006) estimates for probabilities or expectations with respect to the distribution, by taking averages over these sample values.

2.1.2.5 Prediction

It is possible to make predictions by integrating with respect to the posterior:

$$P(newdata|data) = \int_{parameters} P(newdata|parameters)P(parameters|data) \tag{2.6}$$

2.1.2.6 Comparing models

It is possible to compare models based on the marginal likelihood (aka, the evidence) for each model, which is the probability the model assigns to the observed data.

This is the normalizing constant in Bayes' Rule that previously ignored in the equation 2.3:

$$P(data|M_1) = \int_{parameters} P(data|parameters, M_1)P(parameters|M_1) \tag{2.7}$$

Here, M_1 represents the condition that model M_1 is the correct one. Similarly, it is possible to compute $P(data|M_2)$, for some other model that may have different parameter space. One might prefer the model that gives higher probability to the data, or average predictions from both models with weights based on their marginal likelihood, multiplied by any prior preference one have for M_1 versus M_2.

2.1.2.7 Distinctive features of the Bayesian approach

Probability is used not only to describe physical randomness, such as errors in labeling, but also uncertainty regarding the actual values of the parameters. These prior and posterior probabilities represent degrees of belief, before and after seeing the data. A Bayesian model includes fitting a prior distribution for model parameters. If the model/prior are determined without regarding the actual situation, there is no justification for believing the results of Bayesian inference.

The model and prior are selected based on our knowledge of the problem. These choices are not, in theory, affected by the amount of data collected, or by the question one is interested in answering. One do not, for instance, restrict the complexity of the model just because of having only a small amount of data. Pragmatic compromises are inevitable in practice, e.g., no model and prior express the knowledge of the situation. The Bayesian approach relies on reducing such flaws to a level where they will not seriously affect the results. If this is not likely, it may be better to use other approaches.

2.1.3 Bayesian classification models

There are different Bayesian machine learning algorithms in the literature.

In this study, only the following Bayesian models will be used in the proposed prediction model that are explained in greater detail in the following sub-sections:

- Naïve Bayes
- Bayesian Network

2.1.3.1 Naïve Bayes

Naïve Bayes classification models are statistical models that can predict the class label probabilities, such as the probability that an iOS application belongs to the Positive user rating class.

Naïve Bayesian classifiers assume that the effect of criterion value on a given class is independent of the values of the other criteria. This assumption is called the class-conditional independence. It simplifies the computations and in this sense, is considered "naïve". If the assumption holds true for a particular dataset, Naïve Bayes model is the most accurate classification model compare to other classification models but in some datasets there are dependencies between the criteria.

Studies distinguishing classification algorithms have found a simple Bayesian classification model known as the Naïve Bayesian classification model to be comparable in performance with decision tree and selected neural network classification models. Bayesian classification models have also exhibited high accuracy and speed when applied to large datasets (Han *et al.*, 2011).

Given an instance $X = (x_1, ..., x_n)$ to be classified, Naïve Bayesian classifier assigns probabilities to this instance for each of K possible classes: $p(C_k|x_1, ..., x_n)$. In this study, $(x_1, ..., x_n)$ presents the results of evaluation for criteria, and Positive user rating and Negative user rating are two possible classes.

If the number of variables (n) is large then basing such a model on probability tables is not feasible. Therefore, the model can be formulated using Bayes' theorem as the following equation to make it easily computable.

$$p(C_k|X) = \frac{p(C_k)p(X|C_k)}{p(X)} \tag{2.8}$$

$p(X)$, in other words "evidence" does not depend on C and the values of the variables V_i are given, so the denominator is effectively constant. Therefore, the numerator equals the joint probability by considering the equation 2.2. The joint probability can be written as follows:

$$\begin{aligned} p(C_k, x_1, ..., x_n) &= p(C_k)p(x_1, ..., x_n|C_k) \\ &= p(C_k)p(x_1|C_k)p(x_2, ..., x_n|C_k, x_1) \\ &= p(C_k)p(x_1|C_k)p(x_2|C_k, x_1)..p(x_n|C_k, x_1, ..., x_{n-1}) \end{aligned} \tag{2.9}$$

Assuming each variable v_i is conditionally independent of every other variable v_j while $j \neq i$, given the class C:

$$\begin{aligned} p(x_i|C_k, x_j) &= p(x_i|C_k) \\ p(x_i|C_k, x_j, x_k) &= p(x_i|C_k) \end{aligned} \tag{2.10}$$

and so on, for $i \neq j, k$. Thus, the joint model can be formulated as the following:

$$\begin{aligned} p(C_k|x_1, ..., x_n) &\propto p(C_k, x_1, ..., x_n) \\ &\propto p(C_k)p(x_1|C_k)p(x_2|C_k)... \\ &\propto p(C_k)\prod_{i=1}^{n} p(x_i|C_k) \end{aligned} \tag{2.11}$$

Assuming Naïve conditional independence, the conditional distribution over the class C is:

$$p(C_k|x_1, ..., x_n) = \frac{1}{Z}p(C_k)\prod_{i=1}^{n} p(x_i|C_k) \tag{2.12}$$

where the evidence $Z = p(x)$ is a scaling factor dependent only on $x_1, ..., x_n$, that is, a constant if the value of the variables is known.

The Naïve Bayes classifier combines the Naïve Bayes probability model aforementioned above with a decision rule (e.g. pick the hypothesis that is most probable, in other words, the maximum posteriori decision rule). Thus a Naïve Bayes classifier is the function that assigns a class $y = C_k$ for some k as the following:

$$y = argmax_{k \in [k_1, ..., k_n]} p(C_k) \prod_{i=1}^{n} p(x_i | C_k) \tag{2.13}$$

2.1.3.2 Bayesian Network

A Bayesian Network is a graphical model that represents the conditional dependencies among subsets of variables via a directed acyclic graph. Bayesian Networks can be used for classification. Unlike the Naïve Bayes classification model that assumes the class-conditional independence, Bayesian Networks specify joint conditional probability distributions. A Bayesian network is defined by following two components:

- A directed acyclic graph
- A set of conditional probability tables

Each node in the directed acyclic graph represents a random variable in the Bayesian sense. The variables may be observable quantities, latent variables, unknown parameters or hypotheses. The variables can be discrete or continuous-valued. Edges of the graph represent conditional dependencies. Nodes that are not connected to the other nodes represent variables that are conditionally independent of each other.

A Bayesian Network has a conditional probability table for each node. Each node is associated with a probability function that takes, as input, a particular set of values for the node's parent

variables, and gives as output the probability or probability distribution of the variable represented by the node (Han *et al.*, 2011). A Bayesian Network has the following parameters to setup:

- Estimator: An estimator algorithm is required to find the conditional probability tables.
- Search Algorithm: A search algorithm is needed for searching Bayesian Network structures and learn them. The search algorithm typically needs the following to be setup:
  - Initial Network: The initial network is used for structure learning.
  - Markov Blanket Classifier: To ensure that all nodes in the network are part of the Markov blanket of the classifier node, a Markov blanket correction can be applied to learned network structure.
  - Maximum Number of Parents: It is possible to set the maximum number of parents a node can have.
  - Order on Nodes: The order of nodes can be defined as random, or the dataset order can be used.
  - Score Type: There are different indicators to assess the quality of a network structure. Bayes, Akaike Information Criterion (AIC) (Akaike, 1974) and Entropy (Giffin and Caticha, 2007) are some of them.

2.2 Neural Networks

A neural network is a set of connected input/output units in which each connection has a associated weight. Throughout the learning phase, the network learns by tuning the weights to be able to predict the correct class label of the input data.

Neural networks involve long training times and are, therefore, more proper for applications where this is reasonable. They need a number of parameters that are typically thoroughly determined empirically such as the network topology or "structure." Neural networks are often

criticized for their weak interpretability. For instance, it is difficult for humans to interpret the symbolic meaning of the learned weights and of "hidden units" in the network.

Benefits of Neural Networks, however, include their high tolerance of noisy data and their ability to classify the patterns that they have not been trained. They can be employed when there is a limited knowledge of the relationships between variables and classes. Neural Networks are well suited for continuous-valued inputs and outputs, unlike most decision tree algorithms (Han *et al.*, 2011).

2.2.1 MLP (Multi-Layer Perceptron)

The Multi-Layer Perceptron (*MLP*) is a nonlinear feed-forward Artificial Neural Network (*ANN*) which can be applied to regression, classification, and time-series forecasting. Many simple perceptron-like models in hierarchical structure can be shaped a MLP (Witten *et al.*, 2011; Collobert and Bengio, 2004). A multi-layer perceptron has the following parameters to setup:

- Number of Hidden Layers in the network.
- Learning Rate: The amount that weights are updated.
- Momentum: The momentum applied to the weights during updating.
- Seed: Random numbers are used for setting the initial weights of the connections between nodes, and also for shuffling the training data. Seed is used to initialize the random number generator.
- Training Time: The number of epochs to train through.
- Validation Set Size: The percentage size of the validation set.

2.2.2 RBF (Radial Basis Function)

A *RBF* is a real-valued function whose value depends only on the distance from the origin, so that $f(\mathbf{x}) = f(\|\mathbf{x}\|)$; or alternatively on the distance from some other point c, called a center, so that $f(\mathbf{x}, \mathbf{c}) = f(\|\mathbf{x} - \mathbf{c}\|)$. Any function f that satisfies the property $f(\mathbf{x}) = f(\|\mathbf{x}\|)$ is a radial function. The norm is usually the Euclidean distance, although other distance functions are also possible (Witten *et al.*, 2011). The RBF uses the k-means clustering algorithm to provide the basis function and learns a logistic regression on top of that. The following parameters needed to be set for the RBF:

- Clustering Seed: The random seed to pass on to K-means
- Minimum Standard Deviation
- Number of Clusters
- Ridge.

2.3 SVM (Support Vector Machines)

Support Vector Machines (SVM) are supervised learning models with associated learning algorithms that interpret data and determine patterns. SVM can be applied to regression and classification problems. Given a training dataset, each instance labeled as belonging to one of the two classes (TRUE Class: Positive user rating and FALSE Class: Negative user rating class), SVM training algorithm creates a model that assigns new instances to one class or the other.

A SVM model is a representation of the instances as points in the space, mapped so that the instances of the different classes are divided by a perceivable gap that is as wide as possible. New instances are then mapped into that related space and predicted to belong to a class based on the side of the gap they fall on.

In addition to performing linear classification, SVM can efficiently perform a non-linear classification using kernels, implicitly mapping their inputs into high-dimensional feature (In this study, criteria) spaces (Platt, 1998). The following parameters needed to be set for the SVM:

- Complexity parameter
- Kernel

2.4 Conclusion on Machine Learning models

There are many different Machine Learning classification models in the literature. Reviewing all of them is out of the scope of this study. Therefore, three types of Machine Learning classification models have benn briefly reviewed. These three types include Bayesian, Neural Networks, and Support Vector Machines.

The Bayesian classification models are probabilistic models that take variable probabilities into the account. While Naïve Bayes assumes the class-conditional independence, Bayesian Network specifies joint conditional probability distribution.

The Neural Network models are a set of connected input/output units with associated weight for each connection. Neural network learning takes long times because they require a number of parameters that are empirically determined. MLP (Multi-Layer Perceptron) model is a feed-forward Artificial Neural Network model which can be utilized for solving classification problems. On the other hand, RBF (Radial Basis Function) is a real-valued function whose value depends on the distance (e.g. Euclidian distance) from the origin.

The SVM (Support Vector Machines) model is a representation of the mapped instances as points in the space. The gap between instances of the classes divides the classes. The SVMs perform well in high dimensional datasets.

Because of different characteristics of machine learning classification models each can produce different results with different accuracies. In practice, there should be a comparison methodology to select the appropriate model.

CHAPTER 3

PROBLEM STATEMENT

Mobile applications are gaining popularity because of their advantages over other platforms such as portability, location awareness, and accessibility. Mobile application markets and business is growing to fulfill the requirements of mobile application users. Growing market and increasing number of mobile applications and developers is increasing the competition. To be successful in market competition, high quality applications are necessary. Hence, individual developers and development companies need to test and evaluate mobile applications in terms of quality before publishing to the market. One of the essential aspects of quality which most affects user satisfaction, is usability. Therefore, developers and development companies may need to test and evaluate mobile applications in terms of usability.

Individual developers and development companies try to predict if their mobile applications will be successful in mobile application markets. Developers and development companies can examine if the application is ready to be published by evaluating the quality of mobile applications, but they will not know if their application will be successful in the market. One of the obvious measures of success for mobile applications is the user given ratings in mobile application markets such as Apple App Store.

Predicting the user rating before submitting the mobile application to market could help developers and development companies to see the results beforehand and if it is necessary improve the quality of their application. Predicting the negativity/positivity of the user rating of an application in the App Store before releasing it to the App Store can be a classification problem because Apple App Store iOS applications can be classified as applications that have Positive user rating (TRUE class label) or Negative user rating (FALSE class label).

To construct a prediction model with classification techniques the following artifacts are required (Witten *et al.*, 2011):

- A dataset about the applications that already are in the App Store and are rated by the users.
- A classification model to predict the user rating (Negative/Positive classes) of applications in the Apple App Store.

3.1 Research motivation

The iOS (iPhone and iPad Operating System) developed by Apple is very popular and widely used globally. Indeed, Apple reviews App Store submitted APPs in terms of quality and usability. Developers should satisfy minimum requirements in terms of usability, but users decide if they are satisfied with applications and rate them in the App Store. Finally, users tend to buy applications that have good rating score in the App Store. Reviewing the literature has revealed that, there is no study on APP usability evaluation that considers the iOS platform characteristics and embedded application user experiences.

Furthermore, during the literature review it was discovered that none of the mobile application usability evaluation studies consider Apple App Store user ratings and there is not any user rating prediction model which mobile application development individuals and companies can employ to succeed in the mobile application market competition.

3.2 Research objectives

The objective of this research is to explore the following hypotheses:

- H0a - There is not any relationship between the usability of an application and its user given ratings in the Apple App Store.
- H1a - Usability of an iOS application and user given ratings in the Apple App Store are related.
- H0b - It is not possible to predict an iOS application's user given ratings by evaluating its usability and constructing a prediction model.

- H1b - It is possible to predict an iOS application's user given ratings by evaluating its usability and constructing a prediction model.

Therefore, to investigate the research hypotheses, this research study will propose the following artifacts:

- A model to evaluate iOS application usability, and
- A model to predict user ratings of iOS applications in the App Store

3.3 Target audiences

The targeted audiences for this research are the following:

- Application design and development professionals
- User experience design professionals and usability evaluation experts
- Software engineering researchers
- Human-Computer Interaction researchers
- Artificial intelligence and machine learning researchers

CHAPTER 4

RESEARCH METHODOLOGY

This research study explores the hypotheses and approaches the iOS application user rating prediction problem by executing the following phases:

- Phase 1: Identification of required artifacts related to mobile application usability evaluation
- Phase 2: Construction of an iOS application usability evaluation model
  - Expert-based iOS application usability evaluation criteria
  - A User-based iOS application usability evaluation questionnaire
  - Apple App Store user rating evaluation.
- Phase 3: Experiments on iOS application usability evaluation
- Phase 4: Design of a prediction model for the user rating of Apple App Store iOS applications
- Phase 5: Experiments on the machine learning models to predict the user rating of Apple App Store iOS applications

This research study begins by phase 1, identifying the required artifacts related to mobile usability evaluation and specifically APPs. Identified artifacts will be the inputs of upcoming phases. Phase 2 proposes an iOS application usability evaluation model including, expert-based and user-based evaluation methods and Apple App Store iOS application user rating evaluation method. Phase 3 of this research study will use the evaluation model proposed in phase 2 and experiment on a number of applications from the Apple App Store. Phase 4 of this research study will design a prediction model for the Apple App Store user rating taking the data gathered in phase 3 into the consideration. Finally phase 5, will experiment on the

prediction model for the user rating of Apple App Store iOS applications and will select the best resulted model.

Figure 4.1 presents an overview of the research methodology including the inputs, phases, outputs and outcomes. The following subsections in this chapter will provide more details about each phase of this research.

4.1 Identification of required artifacts related to mobile usability evaluation

This first phase of study defines the mobile usability evaluation and covers the identification of required artifacts related to mobile usability evaluation:

- Definition of mobile usability evaluation
- Guidelines to consider during application design
- ISO and literature defined usability measures

  The definition of mobile usability and its evaluation methods, guidelines to consider during application design, and ISO and literature defined measures will be used in defining the criteria of an iOS application usability evaluation model.

- Common usability study scenarios

  The common usability study scenarios will be used in experimentation of iOS application usability evaluation.

This phase takes the following artifacts as inputs:

- ISO Standards
  - ISO 9241 - Ergonomic requirements for office work with visual display terminals
  - ISO 25010 - Systems and software Product Quality Requirements and Evaluation (SQuaRE)

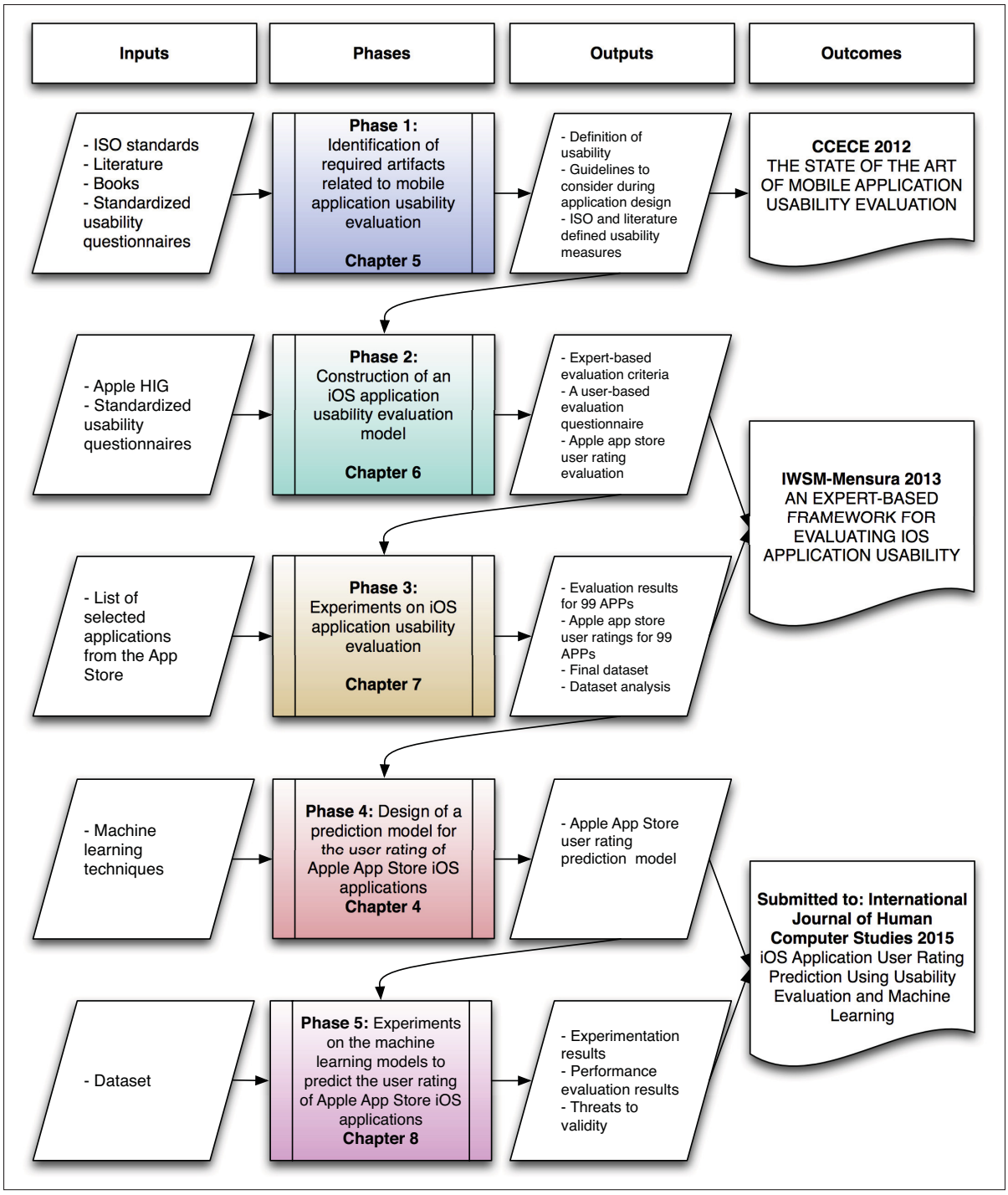

Figure 4.1 Overview of the research methodology

- Literature related to mobile application usability evaluation
- Related books

- Standardized usability questionnaires

Chapter 5 of this research study presents all the above aforementioned artifacts.

4.2 Construction of an iOS application usability evaluation model

This phase of research study aims to provide a model and related criteria for usability evaluation of iOS applications including the following artifacts:

- Expert-based iOS application usability criteria for the use of usability evaluation experts
- A user-based iOS application usability questionnaire
- Apple App Store user rating evaluation

This phase takes the following artifacts as inputs for model construction:

- Apple human interface guidelines (iOS HIG): iOS HIG cover user experience and human interface guidelines specifically for iOS application thus: it is a vital source in construction of a usability evaluation model for APPs.
- Guidelines to consider during application design: There are different usability heuristics, design guidelines and recommendations in the literature that can be used to construct the iOS usability evaluation model.
- ISO and literature defined usability measures: Related ISO standards and some of research studies already defined usability measures that can be used in the expert-based usability evaluation criteria identification and definition,
- Standardized usability questionnaires: Proposed standardized usability questionnaires in the literature can be used to identify the user-based evaluation criteria.

Chapter 6 of this research study explains the iOS application usability evaluation model and its criteria in greater detail.

4.3 Experiments on iOS application usability evaluation

This phase explains the process and results of experiments on iOS application usability evaluation conducted by three researchers utilizing the following sources:

- Expert-based usability evaluation model proposed in this research study
- A subset of 99 selected applications from the App Store for the experimentation

Three researchers (Fatih Nayebi, Prof. Jean-Marc Desharnais and Gustavo Adolfo Vasquez, M.Sc.) are experts from an industry point of view. They have the knowledge of mobile application development, usability evaluation, Apple human-interface guidelines and guidelines to consider during application design.

Each of the three researchers will use the evaluation model and evaluate 99 applications. Next, the researchers will review the results together, discuss and agree on one number for each criterion of each application. This phase is explained in Chapter 7 of this research study by presenting the usability evaluation results of 99 applications as a dataset and its statistical dataset analysis results utilizing the following statistics:

- Central tendency of criteria: Median and Mode values for each criterion
- Dispersion of criteria: Variance, Standard Deviation, Standard Error of Mean, Skewness, Kurtosis, Range and Percentiles for each criterion

4.4 Design of a prediction model for the user rating of Apple App Store iOS applications

This phase of the research study utilizes different machine learning models to construct a prediction model for the user ratings of the Apple App Store APPs by applying the following steps:

- Selection of datasets for training and testing

- Feature subset selection to eliminate unnecessary criteria
- Execution of a number of prediction models on the dataset to find the most suitable prediction model
- Analysis of model performances
- Comparison of the models and selecting the best model for the dataset

This phase has the following outputs:

- Apple App Store user rating prediction model
- Model evaluation indicators to compare the models

Also, this phase takes into the account the following sources:

- Dataset including Apple App Store user ratings
- Machine learning techniques.

The following subsections explain the steps in constructing the prediction model. Also Figure 4.2 presents the overview of the prediction model steps.

4.4.1 Selection of datasets for training and testing

To assess how accurately a prediction model will perform in practice, it is necessary to split the dataset into training and testing datasets. Using training dataset to train a model and then evaluate the accuracy of the resulting learned model on training dataset can guide to misleading overoptimistic estimates due to over-specialization of the learning model to the dataset. Instead, it is better to evaluate the classifier's accuracy on a test dataset consisting of class-labeled instances that were not used to train the model.

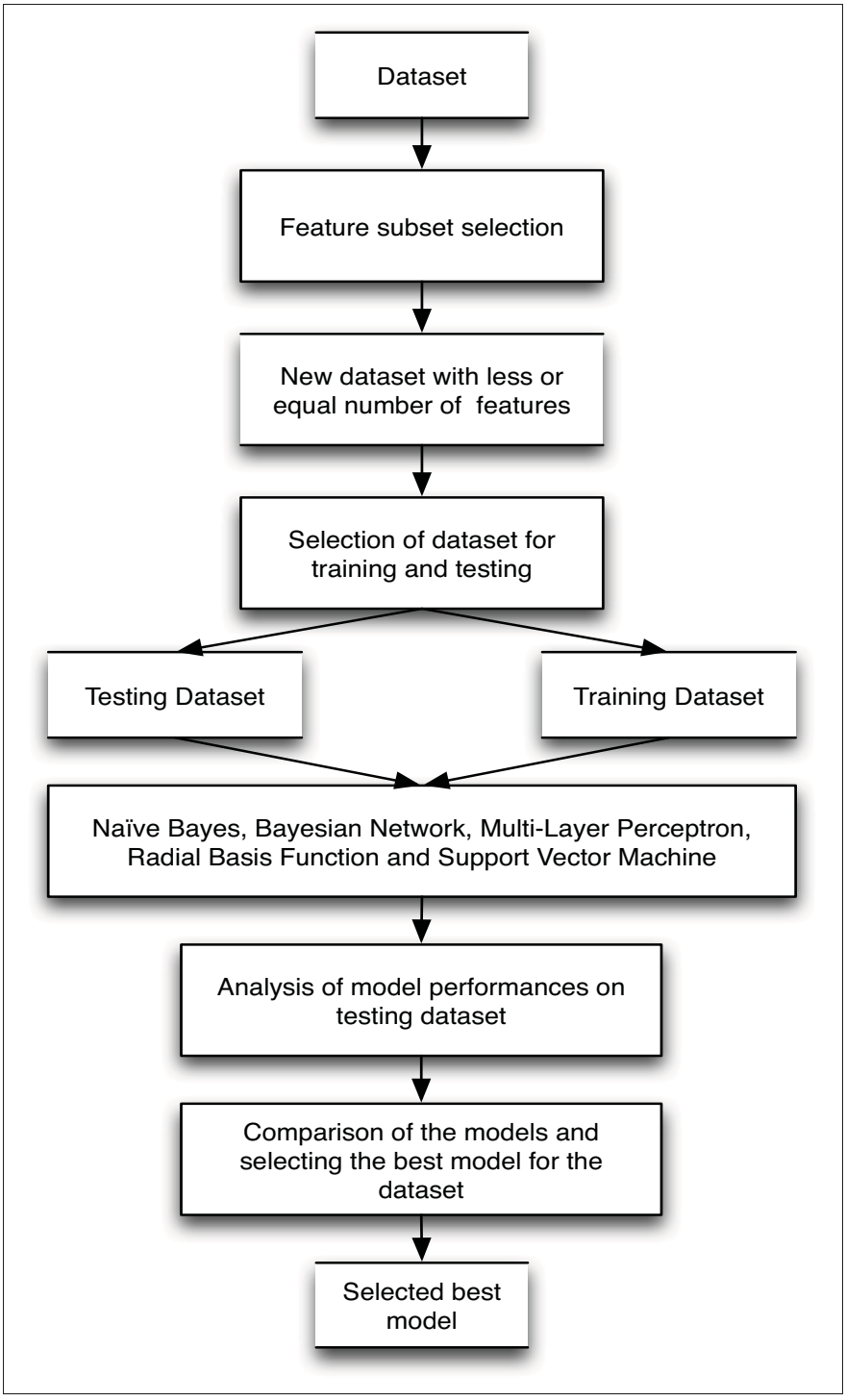

Figure 4.2 Overview of the prediction model

This study will split the dataset into 75% (Train) and 25% (Test) datasets. The dataset has 99 instances (in this study, applications) therefore this phase will provide a training dataset with 74 instances and a testing dataset with 25 instances.

Prediction model will use the training dataset to train itself and will use the testing dataset to test its performance. This approach will avoid problems such as overfitting and will generalize the model to different independent datasets because the training and testing datasets are different.

4.4.2 Feature subset selection

The importance of feature subset selection is that it eliminates features (In this study, criteria) that will decrease the accuracy of the prediction model, as well as decreasing the computational cost of the model. One of the most powerful feature subset selection algorithms is Wrapper, which evaluates all combinations of features using two different algorithms:

- The learning algorithm for prediction, and
- The search algorithm for finding the best combination of features, based on the results of the automated prediction techniques tested.

Various search and learning algorithms are used in Wrapper, but, in general, an exhaustive search algorithm and the K-Nearest Neighbor algorithm outperform the others. In this study, Wrapper is implemented with the KNN $k = 3$ and exhaustive search. Exhaustive search gives superior accuracy, but at the cost of high computational effort (Gutlein *et al.*, 2009).

This cost is not a drawback for this proposed selection process, because the selection model does not need to run during prediction. If there are any changes in the training dataset, model will need to run the feature subset selection and save the selected feature subset to be used during the user rating prediction.

4.4.3 Prediction models

The following subsections present the machine learning classification models and their parameters.

4.4.3.1 Naïve Bayes

This research study utilizes the simple Naïve Bayesian classification model (Witten *et al.*, 2011) with the following parameters:

- Kernel Estimator: A kernel estimator can be used for numeric features rather than a normal distribution. This study does not use kernel estimator because the features are ordinal.
- Supervised Discretization: Supervised discretization can be used to convert numeric features to nominal ones. This study does not require supervised discretization because the features are ordinals.

4.4.3.2 Bayesian Network

Bayesian network classification model uses an estimator and a search algorithm with quality measures. This research study utilizes the Bayesian Network classification model (Bouckaert, 2008) with the following parameters:

- Estimator: An estimator algorithm is required to find the conditional probability tables. This study uses Simple Estimator algorithm for finding the conditional probability tables with alpha 0.5 as the initial count on each value.
- Search Algorithm: A search algorithm is needed for searching Bayesian Network structures and learn them. This study uses local K2 hill climbing algorithm restricted by an order on the features using the following parameters:
    - Initial Network: The initial network that is used for structure learning can be an empty network or a Naïve Bayes network, that is, a network with an arrow from the classifier node to each other node. This study uses Naïve Bayes network as initial network.
    - Markov Blanket Classifier: To ensure that all nodes in the network are part of the Markov blanket of the classifier node, a Markov blanket correction can be applied on learned network structure. This study does not apply the Markov blanket classifier.

- Maximum Number of Parents: It is possible to set the maximum number of parents a node can have. This study sets the maximum number of parents as 1.
- Order on Nodes: The order of nodes can be defined as random or the dataset order can be used. This study uses the dataset order and the class label (TRUE: Positive user rating and FALSE: Negative user rating) comes as the first node.
- Score Type: There are different indicators to assess the quality of a network structure. Bayes, Akaike Information Criterion (AIC) (Akaike, 1974) and Entropy (Giffin and Caticha, 2007) are some of them. This study uses Bayes indicator.

4.4.3.3 MLP (Multi-Layer Perceptron)

Multi-Layer perceptron classification model uses the backpropagation to classify instances. This study employs the MLP (Witten *et al.*, 2011) using the following parameters:

- Hidden Layers: This study uses the hidden layers of the neural network as $(features + classes)/2$.
- Learning Rate: This study applies 0.75 as the learning rate, that is the amount the weights are updated.
- Momentum: This study applies 0.2 as the momentum applied to the weights during updating.
- Nominal To Binary Filter: To improve the performance of the model, this study applies nominal to binary filter to the instances.
- Normalize Features: Normalizing the features can improve the performance of the model. This study normalizes the criteria.
- Seed: Random numbers are used for setting the initial weights of the connections between nodes, and also for shuffling the training data. A seed is used to initialize the random number generator. This study uses seed as 0.

- Training Time: The number of epochs to train through. If the validation set is non-zero then it can terminate the network early. This study uses 1000 as training time.
- Validation Set Size: The percentage size of the validation set. This study sets validation set size as 0 so no validation set will be used and instead the network will train for the 1000 epochs.

4.4.3.4 RBF (Radial Basis Function)

The RBF utilized in this study is a normalized Gaussian radial basis function that uses the k-means clustering algorithm to provide the basis function and learns a logistic regression on top of that. Symmetric multivariate Gaussians are fit to the data from each cluster. It uses the given number of clusters per class (Witten *et al.*, 2011). It standardizes all numeric attributes to zero mean and unit variance. This study employs the RBF with the following parameters:

- Clustering Seed: The random seed to pass on to K-means. This study utilizes a clustering seed with the value of 1.
- Minimum Standard Deviation: This study sets the minimum standard deviation for the clusters as 0.1.
- Number of Clusters: This study sets the number of clusters for K-Means to generate as 2.
- Ridge: This study sets the Ridge value for the logistic regression as 1.0E-8.

4.4.3.5 SVM (Support Vector Machines)

This study uses sequential minimal optimization algorithm for training a support vector classifier (Platt, 1998). This implementation globally replaces all missing values and transforms nominal features into binary ones. It also normalizes all features by default. This study employs the SVM using the following parameters:

- The complexity parameter C is used as 1 in this study.
- This study normalizes the training data.
- Kernel: This study uses the Polynomial Kernel: $K(x,y) = < x,y >^p$.

4.4.4 Model evaluation indicators

To evaluate the performance of the proposed models, a number of indicators will be used in this research study:

- True positives (TP)
- False positives (FP)
- Precision
- Recall
- F-Measure
- Kappa statistic

This section presents the indicators for assessing how good or how "accurate" the model is at predicting the class label of each instance (In this study, application) in the testing dataset.

For each instance, the model's predicted class label is compared with the instance's known class label (In the dataset, positive user ratings are labeled as TRUE and negative user ratings are labeled as FALSE) to examine the accuracy of prediction. The indicators that are used in comparison are given in the following sections.

4.4.4.1 True positives (TP)

The TP refers to the positive instances that are correctly labeled by the classifier. In other words, TP is the number of true positives.

4.4.4.2 True negatives (TN)

The TN refers to the negative instances that are correctly labeled by the classifier. In other words, TN is the number of true negatives.

4.4.4.3 False positives (FP)

The FP refers to the negative instances that are incorrectly labeled as positive. In other words, FP is the number of false positives.

4.4.4.4 False negatives (FN)

The FN refers to the positive instances that are incorrectly labeled as negative. In other words, FN is the number of false negatives.

4.4.4.5 Precision

Precision can be thought of as a measure of exactness (i.e., what percentage of instances labeled as positive are actually such).

$$precision = \frac{TP}{TP+FP} \tag{4.1}$$

4.4.4.6 Recall

Recall is a measure of completeness (what percentage of positive instances are labeled as such). Recall is the same as sensitivity (or the TP).

$$recall = \frac{TP}{TP+FN} = \frac{TP}{P} \tag{4.2}$$

4.4.4.7 F-Measure

An alternative way to use precision and recall is to combine them into a single indicator. This is the approach of the F-measure (also known as the F_1 score or $F-score$) and the F_β measure. They are defined as

$$F = \frac{2 * precision * recall}{precision + recall} \tag{4.3}$$

$$F_\beta = \frac{(1+\beta^2) * precision * recall}{\beta^2 * precision * recall} \tag{4.4}$$

where β is a non-negative real number. The F-measure is the *harmonic* mean of precision and recall (the proof of which is left as an exercise). It gives equal weight to precision and recall. The F_β measure is a weighted measure of precision and recall. It assigns β times as much weight to recall as to precision. Commonly used F_β measures are F_2 (which weights recall twice as much as precision) and $F_{0.5}$ (which weights precision twice as much as recall).

4.4.4.8 Kappa Statistic

Kappa statistic is a chance-corrected measure of agreement between the classifications and the true classes. It is calculated by taking the agreement expected by chance away from the observed agreement and dividing by the maximum possible agreement. A value that is greater than zero means that the model is doing better than chance.

$$kappa = \frac{predictionAccuracy - expectedAccuracy}{1 - expectedAccuracy} \tag{4.5}$$

where, predictionAccuracy is the number of instances that were classified correctly throughout the confusion matrix and expectedAccuracy is the accuracy that any random classifier would be expected to achieve based on the confusion matrix.

There is not a regulated interpretation of the Kappa statistic but Fleiss *et al.* (2004) identify kappas over 0.75 as excellent, 0.40 to 0.75 as fair to good, and below 0.40 as poor.

4.5 Experiments on the prediction model for the user rating of Apple App Store iOS application

This phase will use the following sources:

- Dataset
- Apple App Store iOS application user rating prediction model
- Model evaluation indicators

Moreover, this phase will produce experimentation results and performance evaluation of the results and will select the best model for the dataset. Thus, chapter 8 of this thesis presents the following artifacts for each of five machine learning models:

- Predictions on test dataset:
  - A scatter diagram that presents the predicted class values against actual class values
  - A table which presents the actual, predicted, error, and probability distribution of FALSE and TRUE classes.
- Evaluation on test dataset: A table that presents the correctly/incorrectly predicted instance percentages and the kappa statistic.
- Detailed accuracy by class (TRUE: Positive user rating, FALSE: Negative user rating): A table that presents the following indicators:
  - TP (True Positives)
  - FP (False Positives)
  - Precision
  - Recall
  - F-Measure
- Confusion matrix: A table that presents the prediction model confusion matrix.

CHAPTER 5

IDENTIFICATION OF REQUIRED ARTIFACTS RELATED TO MOBILE USABILITY EVALUATION

This chapter presents phase 1 of the research methodology (See Figure 4.1). This phase has the following inputs:

- Usability related ISO standards
- Usability evaluation related publications in the literature
- Mobile usability related books
- Standardized user-based usability questionnaires

This phase provides the following outputs:

- Definition of usability
- Guidelines to consider during application design
- ISO and literature defined usability measures.

5.1 Usability definition

Usability is a quality characteristic that judges how simple user interfaces (UIs) are to use and learn with effectiveness, efficiency and satisfaction. The UI that a user utilizes can be a mobile application, website, book, tool, machine, process, or anything a human interacts. The term "usability" refers to methods for enhancing ease-of-use through the design process also (ISO/IEC25010, 2011).

Usability includes methods of measuring usability, such as needs analysis and the investigation of the origins of an object's perceived efficiency or style. In human-computer interaction and

computer science, usability studies the simplicity, satisfaction and elegance of interaction with an application.

A usability study may be attended as a principal work function by a usability examiner or as a secondary function by designers, developers, marketing staff, and others.

Different literature studies and international standards define and characterize usability differently. However, the following four aspects of usability for all types of software are considered in the majority of studies.

- Effectiveness: Application effectiveness means, how accurate and complete is an application when users try to achieve goals.
- Efficiency: Application efficiency means, once users have learned the design, how quickly can they perform tasks.
- Learnability: Application learnability means how easy is an application for users to accomplish basic tasks the first time they encounter the design.
- Satisfaction: Application user satisfaction means, how useful, pleasant, comfortable and trusty is it to use the designed application.

5.2 Usability in ISO standards

ISO/IEC9241 (1997) (Ergonomic requirements for office work with visual display terminals) and ISO/IEC25010 (2011) standards are related to usability evaluation and will be explained in more details in following sub-sections.

ISO/IEC9241 (1997) defines usability as “the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use”. Abran *et al.* (2003) consolidate the ISO/IEC9241 (1997), ISO/IEC9126 (2001), ISO/IEC13407 (1999), Dix *et al.* (1993), and Nielsen (1994) usability models, and propose an enhanced model, in other words, a "Consolidated Usability Model". The consolidated

model explains usability as a combination of effectiveness, efficiency, satisfaction, learnability, and security, along with a recommended set of related measures.

More recently, ISO/IEC25010 (2011) breaks down the notion of quality-in-use into usability-in-use, flexibility-in-use, and safety-in-use. Additionally, ISO 25010 defines satisfaction-in-use as:

- Likability: satisfaction of pragmatic goals
- Pleasure: satisfaction of hedonic goals
- Comfort: physical satisfaction
- Trust: satisfaction with security

Also, it defines flexibility-in-use as context conformity-in-use, context extendibility-in-use, and accessibility-in-use.

Figure 5.1 presents a summary of the usability definitions in ISO standards.

5.2.1 ISO 9241 - Ergonomic requirements for office work with visual display terminals

ISO/IEC9241 (1997) presents requirements and recommendations associating the attributes of the hardware, software and environment that contribute to the usability, and the ergonomic principles underlying them. Parts 10 and 12 to 17 of ISO 9241 deal specifically with attributes of the software. More specifically, parts 14-17 are intended to be used by both designers and evaluators of UIs while the focus is primarily towards the designer.

The ISO standards provide an authoritative source of reference, but designers without usability experience have great difficulty employing these types of guidelines. Designers require to know the design aims, advantages of each guideline, the stipulations to apply guidelines, the foundation of the proposed solution, and any procedure that should be followed to implement the guideline successfully.

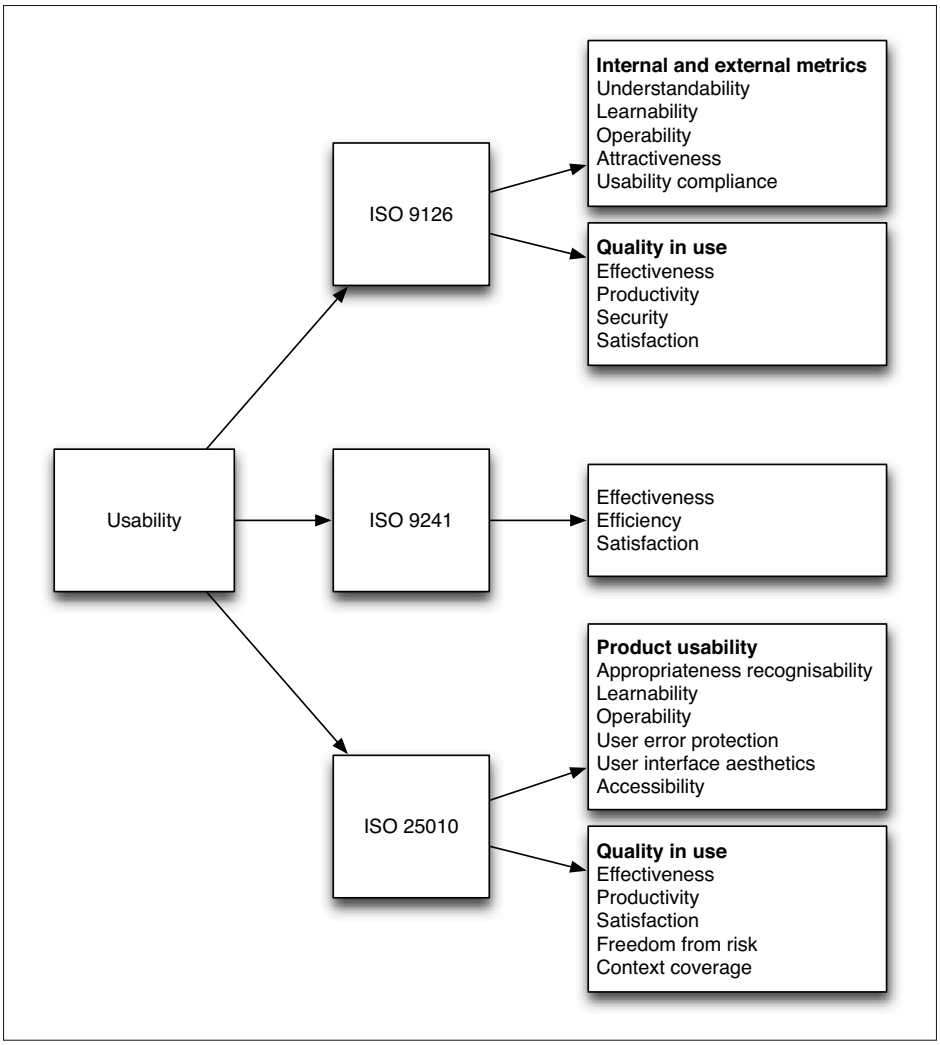

Figure 5.1 Usability definition in ISO standards

Appendix V presents the different parts of ISO 9241 standard.

5.2.2 ISO 25010 - Systems and software Product Quality Requirements and Evaluation (SQuaRE)

ISO/IEC25010 (2011) has three quality models: the quality in use model, the product quality model and the data quality model. ISO/IEC25010 (2011) product quality model defines Usability as "degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use."

Characteristics and sub-characteristics of quality in terms of usability listed in the following subsections.

5.2.2.1 Product quality (usability) characteristics and sub-characteristics

The definition of product quality (usability) in ISO/IEC9241 (1997) is: extent to which an application can be utilized by designated users to accomplish designated goals with effectiveness, efficiency and satisfaction in a particular context of use.

Usability can either be defined or measured as a product quality characteristic in terms of its sub-characteristics or defined or measured directly by measures that are a subset of quality in use.

5.2.2.1.1 Appropriateness recognizability

Appropriateness recognizability means the extent to which users can recognize whether a product or system is appropriate for their needs.

Appropriateness recognizability depends on the ability to recognize the appropriateness of the product or system's functions from initial impressions of the application or system and/or any related documentation.

The information provided by the application can include demonstrations, on-boarding views, tutorials, documentation, or for an iOS application, the information on the Apple App Store.

5.2.2.1.2 Learnability

Learnability means the extent to which an application can be used by designated users to accomplish specified goals of learning to use the application with effectiveness, efficiency, freedom from risk and satisfaction in a specified context of use.

Learnability can be measured or evaluated against the definition above or as the extent to which application properties are corresponding to suitability for learning as defined in ISO 9241-110.

5.2.2.1.3 Operability

Operability means the extent to which an application has attributes that make it easy to operate and control. Operability corresponds to controllability, error tolerance and conformity with user expectations as defined in ISO 9241-110.

5.2.2.1.4 User error protection

User error protection means the extent to which an application protects users against making errors.

5.2.2.1.5 User interface esthetics

User interface esthetics refers to the extent to which the UI of the application enables pleasing and satisfying interaction with the user. This refers to properties of the application that increase the pleasure and satisfaction of the user, such as the use of color and the nature of the graphical design.

5.2.2.1.6 Accessibility

Accessibility means the extent to which an application can be used by people with the widest range of characteristics and capabilities to accomplish a designated goal in a designated context of use.

5.2.2.2 Quality in use characteristics and sub-characteristics

Quality in use is the extent to which an application can be used by specific users to meet their requirements to achieve particular goals with effectiveness, efficiency, freedom from risk and satisfaction in particular contexts of use.

5.2.2.2.1 Effectiveness

Effectivity means the accuracy and completeness with which users achieve specified goals.

5.2.2.2.2 Efficiency

Efficiency refers to the resources expended in relation to the accuracy and completeness with which users achieve goals. Related resources can incorporate time to complete the task (human resources), materials, or the financial cost of usage.

5.2.2.2.3 Satisfaction

Satisfaction means the extent to which user needs are satisfied when an application is used in a designated context of use.

Satisfaction is the response of the user to the interaction with the application and includes attitudes towards the use of the application.

Usefulness: The extent to which a user's perceived achievement of practical objectives, the outcomes of use and the consequences of use satisfy the user needs.

Trust: The extent to which a user or another stakeholder has confidence that an application will behave as intended.

Pleasure: The extent to which a user receives pleasure from fulfilling their personal needs. Personal needs can include needs to acquire new knowledge and skills, to communicate personal identity and to provoke pleasant memories.

Comfort: The extent to which the user is satisfied with physical comfort.

Freedom from risk: The extent to which an application mitigates the potential risk to economic status, human life, health, or the environment.

Economic risk mitigation: The extent to which an application mitigates the likely risk to monetary status, efficient operation, investment property, reliability or other resources in the intended contexts of use.

Health and safety risk mitigation: The extent to which an application mitigates the potential risk to people in the intended contexts of use.

Environmental risk mitigation: The xtent to which an application mitigates the potential risk to property or the environment in the intended contexts of use.

5.2.2.2.4 Context coverage

Context coverage means the extent to which an application can be used with effectiveness, efficiency, freedom from risk and satisfaction in both designated contexts of use and contexts beyond those initially explicitly identified.

Context of use is related to both quality in use and some application quality characteristics and sub-characteristics.

Context completeness: The extent to which an application can be used with effectiveness, efficiency, freedom from risk and satisfaction in all the intended contexts of use.

Context completeness also can be evaluated either as the presence of application properties that aid use in all the intended contexts of use. For instance, the extent to which an application is

usable in different screen size and resolutions, with low internet bandwidth, by a non-expert user; and in a fault-tolerant mode (e.g. no network connectivity).

Flexibility: The extent to which an application can be used with effectiveness, efficiency, freedom from risk and satisfaction in contexts beyond those initially defined in the requirements.

Flexibility enables an application to take account of circumstances, opportunities and individual preferences that may not have been anticipated in advance.

Flexibility can be evaluated either against the above definition or by a capability to be modified to support adaptation to new types of users, tasks and environments, and suitability for individualization as defined in ISO 9241-110.

5.3 Guidelines to consider during application design

Application designers consider distinct guidelines to ameliorate the usability of applications, and usability experts examine applications according to these guidelines to determine the extent to which the design complies with the guidelines. Guidelines can be subjective, and the solutions they provide are contingent to the examiner's judgment. A standardized process and criteria for expert-based usability evaluation based on these heuristics may be beneficial to benchmark and compare different applications.

This section presents UI design guidelines to consider during the application design process. Figure 5.2 lists the guidelines.

In later sub-sections, the guidelines and related criteria to evaluate them are defined.

5.3.1 Usability heuristics

Usability Heuristics are rules of thumb, which UI designers may follow for better designs. Usability examiners can also consider them during usability evaluation. Sets of usability heuristics

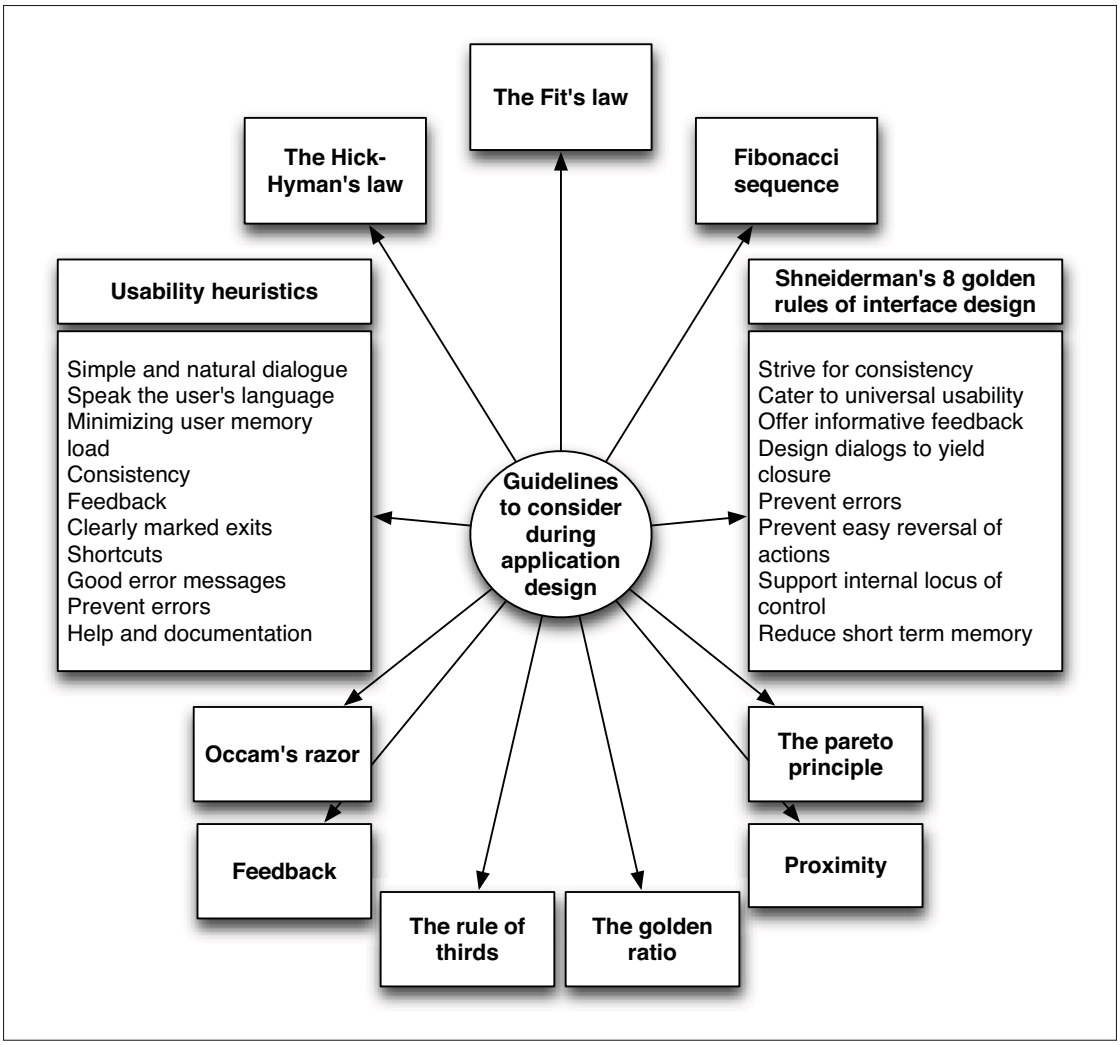

Figure 5.2 Guidelines to consider during application design

have been developed by Nielsen (1994), Gerhardt-Powals (1996), Shneiderman *et al.* (2010), and Weinschenk and Barker (2000) – see Figure 5.3.

This study examines these sets of usability heuristics, and selects a subset of them for the usability evaluation of iOS applications, and defines the criteria to evaluate an app's level of compliance with the heuristics. The responses to the questions are multiple choices and quantified on an ordinal scale (from 1 to 5, in this case).

The following sub-sections include the subset of usability heuristics that will be used in this study.

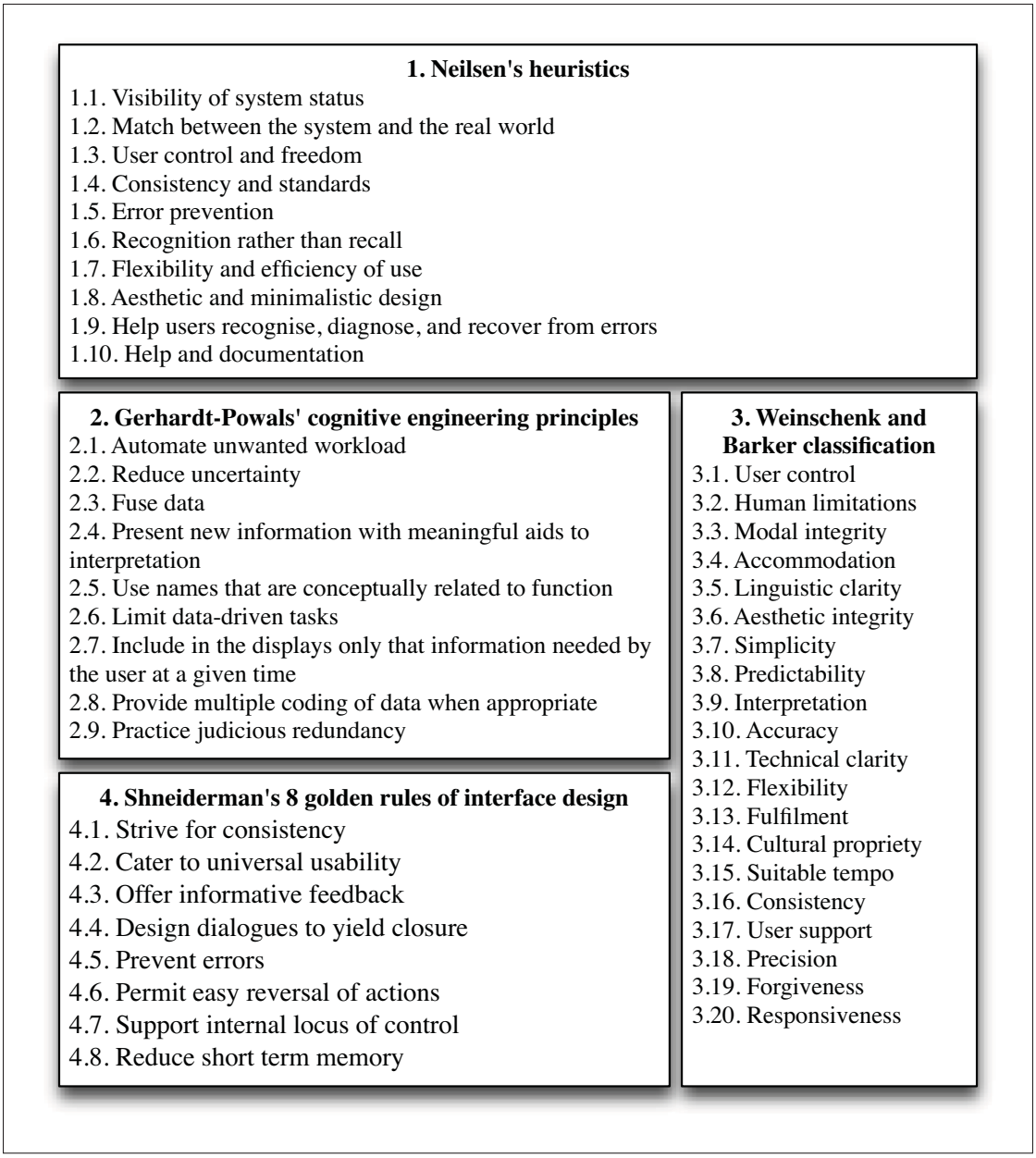

Figure 5.3 Usability heuristics

5.3.1.1 User Control and freedom

The UI provides control and freedom to the user.

a. The user can leave an undesired situation via clearly marked cancel/exit points.

b. The app supports undo and redo.

c. The user can initiate and control the velocity and sequencing of the interaction.

d. The user comprehends where he is in the application, how he got there, and where he can go ahead via a navigation controller stack and accurate view names.

e. The user is held informed of his current position and of the number of steps he must go through to reach his goal.

5.3.1.2 Error correction

Error correction messages should express the problem precisely, in plain language, and suggest a solution.

a. When an error occurs, the application notifies the user about what happened, why, and how to fix it.

b. Required entry fields are made apparent to the user via visual indicators.

c. A back button or gesture returns the app to a previous view without loss of data.

5.3.1.3 Human Limitations

The design takes into account human limitations, both cognitive and sensory, and helps the user via memory support and appropriate instructions.

5.3.1.4 Accommodation

The design meets the needs and behaviors of all targeted user groups and is user-friendly.

a. The language of the application embodies words, phrases, and concepts that are familiar to the user.

b. Appropriate metaphors portraying real-life objects are used to help the user understand and learn, the task.

c. The UI is suitable for the user's task and skill level, empowering him to focus on the task, rather than on the technology chosen to perform the task.

5.3.1.5 Linguistic Clarity

The used language is clear, adequate, and appropriate for the audience.

a. The application contains no spelling or grammatical mistakes, which could damage the user's trust.

b. The application does not use abbreviations and acronyms unless they are straightforward and easily understood.

5.3.1.6 Esthetic integrity

The design is visually pleasing and well integrated into the functionality of the application, providing similarity, continuity, completion, proximity, and figure, image/background recognizability.

a. Similarity transpires when objects appear alike to each other, and one can recognize them as part of a group or pattern.

b. Continuity happens when the eye is compelled to move from one object to another.

c. Completion occurs when a user can perceive an incomplete object or non-filled space as a whole throughout adding the missing information.

d. Proximity occurs when elements are placed adjacently together, and user can perceive them as belongings to a group.

e. Figures; Forms, silhouettes, and shapes are contrasted with a background and the surrounding area. Balancing figures and background can make the perceived image clearer. Unusual figure/background ties can append interest and subtlety to a picture.

5.3.1.7 Simplicity

Simplicity refers to the tailoring of the design to attract and engage the targeted user groups, and to avoid the unnecessary complexity, owing to the restricted screen size. The following criteria aim to evaluate the simplicity.

a. Views comprise information and content that is relevant or needed.

b. Color coding is used for clarity where appropriate.

c. The size of figures is optimized for performance and proper resolution for response time impact

d. The number of colors is restricted to 3-4.

e. The application's purpose and usage area is immediately understandable in the commencement.

5.3.1.8 Predictability

The user will be capable of predicting the behavior of the system in response to his actions.

a. The application clarifies which dialog the user is in, where he is in the application, and which activities he can perform and how to perform them.

5.3.1.9 Flexibility

The design is flexible adequately to adapt to the necessities and habits of the user.

Shortcuts may accelerate the interaction between the user and the application, and may also increase user control over the application.

a. The application permits the user to work in a manner that suits him.

b. The user does not need to use workarounds or manuals.

c. The most frequently used elements of the application are placed from top-left to bottom-right, in order of importance.

d. The user can customize the UI to revise his interaction with the application and to have the information exhibited in a style that suits his capabilities and requirements.

e. Tappable and un-tappable areas of the app are clearly recognizable.

f. Shortcuts have been developed for the most commonly used elements of the application.

g. The navigation points between starting locations and tasks are apparent to the user.

5.3.1.10 Consistency

The styles and behaviors of the different parts of the application are consistent.

Users should not have to think about the meaning of words, states, or actions in an application. Interactions between the user and the application, and the utilization of gestures should be consistent.

a. The UI adheres to the user's expectations, meets the predictable contextual needs of the user and respects accepted conventions.

b. All the views are presented consistently so users can apply knowledge obtained in one part of the application to other parts of the application.

c. Labels and titles are consistent throughout the app, and accurately define the tasks to be performed in the app.

d. The app's interactions and gestures meet the expectations of the user, in that they are standard and predictable.

5.3.1.11 User Support

The design of the application supports learning and gives the required assistance to the user.

It is best if an application can be used without any help or documentation. If it is required by the user, the application should provide a list of concrete steps to accomplish the tasks.

a. The app provides readily available help to users when required.

b. The Help documentation is adequately prepared and is both appropriate and informative.

c. The user can easily move from Help to the current task.

d. Help does not interfere with the task flow.

e. Help is context-based and addresses all the necessary contexts.

5.3.1.12 Forgiveness

The app 'forgives' the user for committing an error, and enables him to recover successfully by providing precisely marked exits.

a. The UI is error-tolerant, with error management tactics in place to deal with errors without the necessity of any further user action.

   A careful design that prevents a problem from occurring in the first place is even better than good error management.

b. The app makes it difficult to make mistakes by preventing them with items like a confirm command.

c. The app validates the information that the user enters into data forms, informing him if it is not in an acceptable format.

5.3.1.13 Responsiveness

The UI is responsive, provides sufficient and timely feedback about the app's status, and signals task completion.

a. The app keeps the user informed about the send/receive status of content via a progress indicator.

b. The UI supports Undo and Redo.

c. The application enables the user to leave an unwanted state without having to embark on an extended UI interaction.

5.3.2 Other guidelines

Appendix VI presents other guidelines to be considered during application design.

5.4 Mobile human interface guidelines

With the emergence and rapid deployment of mobile technologies, a number of additional studies such as Ryu (2005) and Gafni (2009) have focused on the usability of mobile devices: "Problems caused by physical restrictions of mobile devices and wireless networks imply that while designing and conducting usability studies for mobile applications, these issues must be carefully examined in order to select an appropriate research methodology and minimize the potential effect of contextual factors on perceived usability when they are not the focus of studies". For Zhang and Adipat (2005) mobile usability includes some of the new mobility-related challenges, such as Mobile Context, Connectivity, Small Screen Size, Different Display Resolutions, Limited Processing Capability and Power, and Data Entry Methods.

At the same time, mobile device manufacturers have been enforcing their usability constraints. For example, the Apple (2013) iOS Human Interface Guidelines state the iOS platform characteristics that should be considered during the application development process, such as: Interaction with Multi-Touch screen, Displays of different resolutions and dimensions, Device orientation changes and Gestures such as tap, flick, and pinch. In addition, Apple reviews applications submitted for the App Store based on these characteristics.

Concurrently, Google (2014) has developed Android user interface guidelines, which guide developers to take into account the following characteristics: Touch gestures, size and location of Icons and Buttons, Contextual Menus and their responsiveness, simplicity, size, and format of Text, and certain aspects of Messages. These guidelines also explain how these characteristics should be considered during the development and testing of Android applications.

This study is concentrated on iOS application usability evaluation and user rating prediction because:

- iOS and Android are different and should be considered separately in usability studies
- iOS is in the market since the longer time
- There are more applications available
- iOS is specific to Apple hardware
- Apple has published iOS Human Interface Guidelines and reviews applications for conformity.

The Apple (2013) Human Interface Guidelines (HIG) are intended to help designers and developers build the highest quality UIs for their iOS applications and offer the best possible user experience. Apple claims that working with the platform conventions will better position developers to create more user-friendly iOS applications. To fulfill and guide developers through this claim, Apple provides grouped guidelines as follows:

- Great iOS APPs embrace the platform and human interface principles.
  - Platform characteristics.
  - Human interface principles.
- Great APP design begins with some clear definitions.
- A great user experience is rooted in developer' s attention to detail.
  - User experience guidelines.
  - iOS element usage guidelines.
- People expect to find iOS technologies in the APPs they use.
- All APPs need at least some custom artworks.

Apple has developed UI elements and guided developers and designers to use them in appropriate way during the design and development process with the guidance of Apple iOS HIG.

On the other hand, Apple provides "Principles of User Interface Design" and forces developers to develop APPs compliant with it by reviewing APPs submitted to App Store. Hence Apple HIG gives a significant amount of knowledge, importance and guidance for better user experience that should be considered.

In this study the following subsections of HIG are employed for iOS application usability evaluation and user rating prediction:

- Platform Characteristics
- Human Interface Principles
- User Experience Guidelines

Figure 5.4 presents the characteristics of the platform, as well as the human interface principles and the user experience guidelines from the HIG document. We describe and explain the user

experience guidelines that are the most relevant to this study, and propose questions that can be asked for quantifying the level of compliance with the guidelines.

5.4.1 Platform characteristics

iOS-based devices share several unique characteristics that influence the user experience of all APPs that run on them. The most successful applications embrace these features and provide a user experience that integrates with the device. Apple, considers following platform characteristics for user experience perfection:

- Consistency of display with different resolutions
- Device orientation change
- UI gestures
- Single app interactivity and single window
- App preferences in built-in setting app
- Minimalistic onscreen help and being easy to use

The following subsections describe each platform characteristics in greater detail.

5.4.1.1 The display is paramount, regardless of its size

The display of an iOS device is at the heart of the user's experience. Not only do people view beautiful text, graphics, and media on the display, they also physically interact with the Multi-Touch screen to drive their experience (even when they can't see the screen).

Different iOS devices can have displays of various dimensions and resolutions, but in all devices the display affects the user experience in the same ways:

- 44 x 44 points is the comfortable minimum size of a tappable UI element.

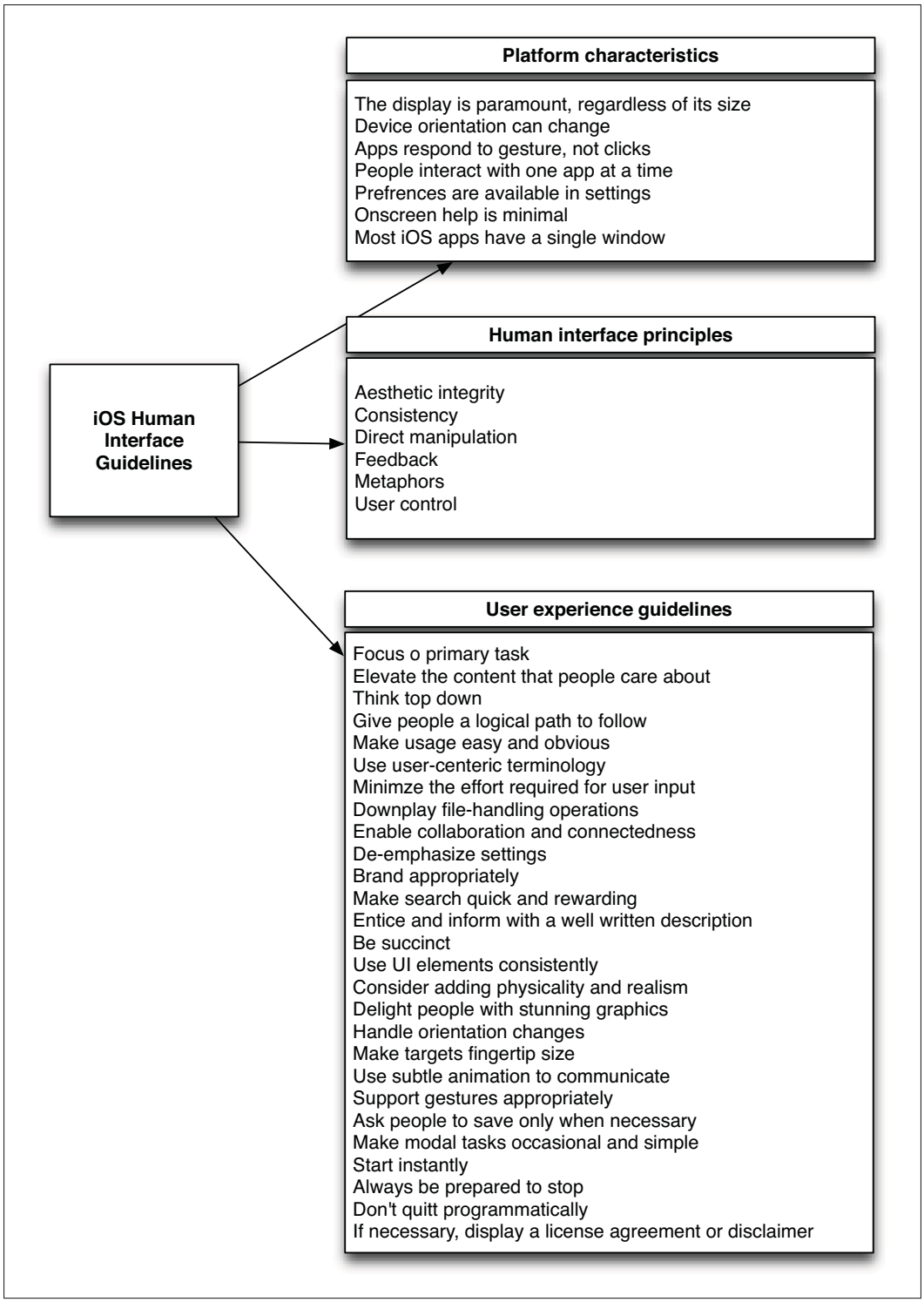

Figure 5.4 iOS Human Interface Guidelines and usability evaluation

- People are very aware of the quality of app artwork.
- The display encourages people to forget about the device and to focus on their content or task.

This characteristic insists on "Consistency of display with different resolutions" and is related to "Consistency" in Human Interface Principles.

5.4.1.2 Device orientation can change

People can rotate iOS devices at any time and for a variety of reasons. For example, sometimes the task people are performing feels more natural in portrait, and sometimes people feel that they can see more in landscape. Whatever their reason for rotating the device, people expect the app to maintain its focus on the primary functionality.

People often launch APPs from the Home screen, so they tend to expect all APPs to start with the same orientation.

5.4.1.3 APPs respond to gestures, not clicks

People make particular finger movements, called gestures, to operate the unique Multi-Touch interface of iOS devices. For example, people tap a button to activate it, flick or drag to scroll a long list, or pinch open to zoom in on an image.

The Multi-Touch interface gives people a sense of immediate connection with their devices and enhances their sense of direct manipulation of on-screen objects.

People are comfortable with the standard gestures because the built-in APPs use them consistently. Their experience using the built-in APPs gives people a familiar set of gestures that they expect to be able to use successfully in most other APPs.

5.4.1.4 People interact with one app at a time

Only one app is visible in the foreground at a time. When people switch from one app to another, the previous app transitions to the background and its UI goes away. This feature, called multitasking, allows APPs to remain in the background until users relaunch them or until they are terminated.

Most APPs enter a suspended state when they transition to the background. Suspended APPs are displayed in the multitasking UI, which provides a convenient way for people to switch to recently used APPs.

5.4.1.5 Preferences are available in settings

People set certain preferences for an iOS app in the built-in Settings app. They must switch away from the current app when they want to access those preferences in Settings.

Preferences in the Settings app are of the "set once and rarely change" type. Although some of the built-in APPs provide preferences of this kind, most applications do not need them, so they do not have preferences in the Settings APP.

5.4.1.6 Onscreen user help is minimal

Mobile users have neither the time nor the desire to read through much help content before they can benefit from an app. What's more, help content takes up valuable space to store and display.

iOS devices and the built-in APPs are intuitive and easy to use, so people do not need onscreen help content to tell them how to use the device or the applications. This experience leads people to expect all APPs to be similarly easy to use.

5.4.1.7 Most APPs have a single window

An APP has a single window unless it supports an external display. An application's window fills the device's main screen and provides an empty surface that hosts one or more views to present the content. It is vital to realize that a window in an iOS app is very different from a window in a computer application. For instance, an iOS window has no visible components (such as a title bar or a close button), and it cannot be moved to a new location on the device display.

It is also important to realize that most users are unaware of the windows and views in the APPs that they use. For the most part, users experience an iOS app as a collection of screens through which they navigate. From this perspective, a screen commonly corresponds to a distinct visual state or mode in an app. In the Contacts app on iPhone, for example, users think of their contact list—regardless of its length—as one screen and an individual contact's details as a different screen.

5.4.2 Human interface principles

An excellent UI follows human interface design principles regarding the way users think and work, not the specifications of the device. A UI that is unattractive, convoluted, or illogical can make even a great application hard to use. However, a beautiful, intuitive, compelling UI enhances an application's functionality and stimulates a positive emotional attachment in users. Human Interface Principles proposed by Apple iOS HIG are as following:

- Aesthetic Integrity
- Consistency
- Direct Manipulation
- Feedback
- Metaphors

- User Control

5.4.2.1 Esthetic integrity

Esthetic integrity is not a measure of how beautiful an app is. It is a measure of how well the appearance of the app integrates with its function. For example, an app that enables a productive task keeps decorative elements subtle and in the background, while giving prominence to the task by providing standard controls and behaviors. Such an app gives users a clear, unified message about its purpose and its identity. If, on the other hand, the app enables the productive task within a UI that seems whimsical or frivolous, people might not know how to interpret these contradictory signals.

Similarly, in an app that encourages an immersive task, such as a game, users expect a beautiful appearance that promises fun and encourages discovery. Although people do not expect to accomplish a serious or productive task in a game, they still expect the game's appearance to integrate with the experience.

5.4.2.2 Consistency

Consistency in the interface allows people to transfer their knowledge and skills from one app to another. A consistent app is not a slavish copy of other APPs. Rather, it is an application that takes advantage of the standards and paradigms people are comfortable with.

To determine whether an app follows the principle of consistency, think about these questions:

- Is the app consistent with iOS standards? Does it use system-provided controls, views, and icons correctly? Does it incorporate device features in a reliable way?
- Is the application consistent within itself? Does the text use uniform terminology and style? Do the same icons always mean the same thing? Can people predict what will happen when

they perform the same action in different places? Do custom UI elements look and behave the same throughout the app?

- Within reason, is the app consistent with its earlier versions? Have the terms and meanings remained the same? Are the fundamental concepts essentially unchanged?

5.4.2.3 Direct manipulation

When people directly manipulate onscreen objects instead of using separate controls to manipulate them, they are more engaged in the task, and they more readily understand the results of their actions. iOS users enjoy a heightened sense of direct manipulation because of the Multi-Touch interface. Using gestures gives people a greater affinity for, and a sense of control over, the objects they see onscreen because they can touch them without using an intermediary, such as a mouse.

For example, instead of tapping zoom controls, people can use the pinch gestures to expand directly or contract an area of content. Also, in a game, players move and interact directly with onscreen objects. For example, a game might display a combination lock that users can spin to open.

In an iOS app, people can experience direct manipulation when they:

- Rotate or otherwise move the device to affect onscreen objects
- Use gestures to manipulate onscreen objects
- Can see that their actions have immediate, visible results

5.4.2.4 Feedback

Feedback acknowledges people's actions and assures them that processing is occurring. People expect immediate feedback when they operate a control, and they appreciate status updates during lengthy operations.

The built-in APPs respond to every user action with some perceptible change. For example, list items highlight briefly when people tap them. During operations that last more than a few seconds, a control shows elapsing progress, and if appropriate, the app displays an explanatory message.

Subtle animation can give people meaningful feedback that helps clarify the results of their actions. For example, lists can animate the addition of a new row to help people track the change visually.

Sound can also give people useful feedback. However, a sound should not be the primary or sole feedback mechanism because people may use their devices in places where they cannot hear or where they must turn off the sound.

5.4.2.5 Metaphors

When virtual objects and actions in an app are metaphors for objects and actions in the real world, users quickly grasp how to use the app. The classic example of a software metaphor is the folder: People put things in folders in the real world, so they immediately understand the idea of putting files into folders on a computer.

The most appropriate metaphors suggest a usage or experience without enforcing the limitations of the real-world object or action on which they are based. For example, people can fill software folders with much more content than would fit in a physical folder.

In general, metaphors work best when they are not stretched too far. For example, the usability of software folders would decrease if they had to be organized into a virtual filing cabinet.

5.4.2.6 User control

People, not APPs, should initiate and control actions. Although an app can suggest a course of action or warn about dangerous consequences, it is usually a mistake for the app to take

decision-making away from the user. The best APPs find the correct balance between giving people the capabilities they need while helping them avoid dangerous outcomes.

Users feel more in control of an app when behaviors and controls are familiar and predictable. Moreover, when actions are simple and straightforward, users can easily understand and remember them.

People expect to have ample opportunity to cancel an operation before it begins, and they expect to get a chance to confirm their intention to perform a potentially destructive action. Finally, people expect to be able to, gracefully stop an underway operation.

5.4.3 User experience guidelines

Appendix VII presents the Apple HIG user experience guidelines in detail.

5.5 Usability evaluation methods

A usability evaluation method is a procedure which is composed of a set of well-defined activities for collecting usage data related to end-user interaction with a mobile application and/or how the specific properties of this mobile application contribute to achieving a certain degree of usability.

Usability Evaluation Methods were formerly developed specifically to evaluate WIMP (Window, Icon, Menu, Pointing device) interfaces. One of the most representative examples is the heuristic evaluation method proposed by Nielsen (1994). Since mobile UIs have grown in importance, new and adapted evaluation methods have emerged to address this type of UIs.

Not all of usability evaluation methods are used in the mobile usability context. Typically, three types of evaluation methodologies are currently used in mobile usability studies:

- Expert-based evaluation: defined aspects of applications are evaluated directly by experts to evaluate the usability.

- User-based evaluation: users are provided with applications and asked about their experience.
- Laboratory experiments: human participants are required to perform specific tasks using a mobile app in a controlled laboratory setting.

There are two general types of usability evaluation methods in terms of who performs the evaluation. First type is usability evaluation performing by experts, and the other type is usability evaluation performing by the users of the application.

5.5.1 Expert-based evaluation

Expert-based evaluations are primarily structured inspections by APP usability experts. An APP usability evaluation expert is a professional with the iOS APP development, user experience design and usability evaluation knowledge and experience. Expert-based methods refer to any form of usability evaluation that involves an expert examining the application and evaluating its likely usability for a given user population. In such cases, users are not employed, and the basis for the evaluation lies in the interpretation and judgment of the examiner. There is considerable interest in this form of evaluation since it can produce results faster and presumably cheaper than user-based evaluations (Zhang and Adipat, 2005; Duh and Tan, 2006).

Expert-based evaluation is often used in conjunction with user-based evaluation and always come before user-based evaluation. Usability evaluation experts are experts in interfaces, but they are typically not experts in the tasks to be performed on a particular interface. Conversely, users are typically experts in performing the tasks but are not experts in interface design.

Expert-based evaluation methods typically point to heuristic evaluation methods: guidelines to consider for designing usable applications and cognitive biases in mobile usability evaluation domain.

Heuristic evaluation is done by looking at an interface and trying to come up with an opinion about what is good and bad about the interface. Ideally people would conduct such evalua-

tions according to certain rules, such as those listed in typical guideline documents by Nielsen (1994). Apple (2013) iOS human interface guidelines document can be considered during heuristic usability evaluation for iOS application usability evaluation.

Characteristics of expert-based evaluation methods:

- Takes place in early design stage of iterative design
- Does not need users
- Finds individual usability problems and can address expert user issues
- Does not involve real users so does not find surprises relating to their needs

Gafni (2009), for instance, in a study of mobile wireless information systems, has developed mobile device-specific usability measures, such as: display load, clarity of operation possibilities, completeness of operation menu, and display self-adjustment possibilities, their purpose and method of calculation. In addition, Gafni links these measures to three types of wireless mobile-related problems: network, device, and mobility.

Hussain and Kutar (2009) define a usability metric framework for mobile phone applications and use the Goal Question Metric approach to link usability goals, such as simplicity, accuracy, and safety, to questions and related metrics.

5.5.2 User-based evaluation

User-based evaluation methods are methods where a user is involved in the evaluation. In user-based evaluation methods, a sample of users perform a set of pre-determined tasks to examine the extent to which the application supports the intended users.

Tightly coupled to the operational approach to usability definition, the user-based evaluation methods draw heavily on the experimental design tradition of human factors psychology in

employing task analysis, pre-determined dependent variables and, usually, quantitative analysis of performance supplemented with qualitative methods.

In a typical user-based evaluation, users are asked to perform a set of tasks with the application. After the tasks are completed, users are often asked to provide data on likes and dislikes through a questionnaire. One can study many aspects of usability by just asking the users. Aforementioned is especially true for issues relating to the users' subjective satisfaction and possible anxieties, which are hard to measure objectively. Questionnaires and interviews are also useful methods for studying how users use the application and what features they particularly like or dislike. Questionnaires and interviews are direct methods when it comes to measuring user satisfaction.

There are many standardized usability questionnaires that is used in the literature and can be examined for mobile application usability evaluation. Questionnaires are listed and explained in more detail in Appendix VIII.

Characteristics of user-based evaluation methods:

- Take place in task analysis and follow-up studies
- Need users
- Find subjective user preferences, are easy to repeat, probe in-depth attitudes and experiences
- Need pilot work to prevent misunderstandings that are time-consuming and hard to analyze and compare.

5.5.3 Laboratory experiments

Usability laboratory experiments take place in a usability lab where the evaluation takes place. It is an environment in which users are studied while they interact with a mobile application to evaluate its usability.

By controlling the environment and giving predefined tasks to the users in a usability experiment, it is possible to ensure that they test all aspects of usability. The environment can also be controlled in such a way to isolate users from conditions that are not relevant to the experiment. These advantages make lab experiments useful for comparing different mobile designs and interfaces.

However, isolating users from environmental factors that can affect usability may cause differences in the user experience, and the effect of environmental factors prevalent in the real world may not be felt. According to Zhang and Adipat (2005) it is also reported that organizing lab experiments is more costly than other methodologies, because of the equipment required.

5.5.4 Comparison of methods

In the mobile domain, the above considered methods have several advantages and disadvantages. Since the majority of mobile applications are developed for many different end-user profiles, user-based evaluations such as laboratory experiments and user-based evaluation methods can take into account a wide range of end-users. However, the use of these methods may not be cost-effective since they require many resources. These methods also need a full or partial implementation of the mobile application, signifying that usability evaluations are mainly moved to the last stages of the mobile application development process.

Expert-based evaluation methods, on the other hand, allow usability evaluations to be performed on mobile artifacts such as mock-ups, UI controls, paper prototypes, or UI models. This is relevant because these artifacts can be created during the early stages of the mobile application development process. Another benefit of the expert-based evaluation is that it often requires fewer resources than other methods. However, the usability evaluation performed may be limited by the quality of the guidelines or evaluator expectations.

5.6 Common usability study scenarios

It is important to consider the issues when choosing criteria for a usability study, including the goals of the study, the user, the technology that is available to collect and analyze the data, and the budget and time to resolve findings. Every usability study has advantages and disadvantages, and it is not possible to prescribe the exact criteria to use for every type of usability evaluation studies. Nielsen (1994) and Tullis and Albert (2013), identify categories of usability studies and develop recommendations about criteria.

Appendix IX explains common usability study scenarios according to Tullis and Albert (2013) and Nielsen (1994).

5.7 Standardized usability questionnaires

This section provides standardized usability questionnaires collected and described by Sauro and R.Lewis (2012) in four categories as follows:

- Post-study questionnaires
- Post-task questionnaires
- Assessing perceived usability of websites questionnaires
- Other questionnaires

Appendix VIII presents the standardized usability questionnaires in detail.

Different standardized user-based questionnaires can be summarized as follows:

- Standardized questionnaires fall into four varying categories: post-study, post-task, website, and other.
- There is not any standardized questionnaires for assessing perceived usability of mobile applications

- Standardized post-study questionnaires include the QUIS (Questionnaire for User Interface Satisfaction), SUMI (Software Usability Measurement Inventory), PSSUQ (Post-study System Usability Questionnaire), SUS (Software Usability Scale), USE (Usefulness, Satisfaction, and Ease of Use), and UMUX (Usability Metric for User Experience).

- Standardized post-task questionnaires include the ASQ (After-scenario Questionnaire), ER (Expectation Ratings), SEQ (Single Ease Question), SMEQ (Subjective Mental Effort Question), and UME (Usability Magnitude Estimation).

- All of the post-study and post-task questionnaires are of potential value to usability evaluators because of psychometric qualification indicating significant reliability, validity, and sensitivity.

- Head-to-head comparisons of the methods indicate that the most sensitive post-study questionnaire is the SUS, followed by the PSSUQ; the most sensitive post-task questionnaire is the SMEQ, followed by the SEQ.

- Due to their growing use for commercial transactions, standardized usability questionnaires for websites include items focused on the assessment of attributes such as trust and service quality.

- The scores from standardized usability measurements do not have any inherent meaning, but they are useful for comparisons, either between applications or conditions in usability studies or against databases.

- Commercial usability questionnaires that provide comparison with related databases are the SUMI, WAMMI (Website Analysis and Measurement Inventory) and SUPR-Q (Standardized Universal Percentile Rank Questionnaire).

- For non-commercial usability questionnaires, data in the public domain is available for the PSSUQ and CSUQ.

- Questionnaires from the market research literature that may be of interest to APP usability evaluators are the ASCI (American Customer Satisfaction Index), NPS (Net Promoter

Score), CxPi (Forrester Customer Experience Index) and TAM (Technology Acceptance Model) scales (Perceived Usefulness and Perceived Ease of Use).

5.8 Usability measures

Usability measures should be considered in usability evaluation methods. This section lists and describes the candidate usability measures for common usability study scenarios. Furthermore, Figure 5.5 presents usability measures proposed in ISO standards and literature as well.

5.8.1 ISO defined usability measures

This section explains the ISO defined usability measures.

5.8.1.1 Effectiveness

Effectiveness is the first quality in use characteristic in ISO 9126-4. The considered measures are listed next.

Task effectiveness What proportion of the goals of the task is achieved correctly?

Task completion What proportion of the tasks is completed?

Error frequency What is the frequency of errors?

5.8.1.2 Productivity

Productivity is the second quality in use characteristic in ISO 9126-4. The considered measures are listed next.

Task time: How long does it take to complete a task?

Task efficiency: How efficient are the users?

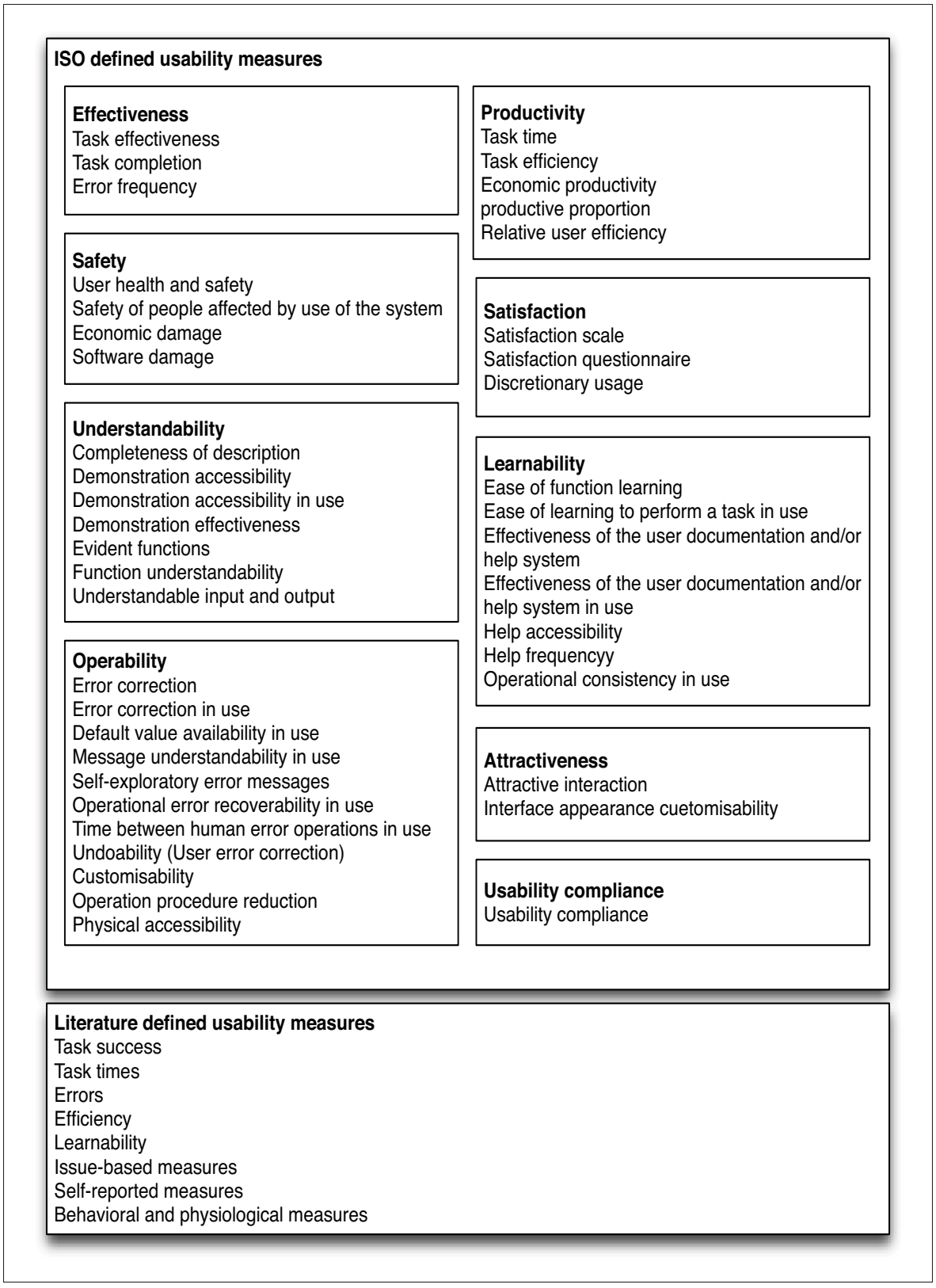

Figure 5.5 Usability measures

Economic productivity: How cost-effective is the user?

Productive proportion: What proportion of the time is the user performing productive actions?

Relative user efficiency: How efficient is a user compared to an expert?

5.8.1.3 Safety

Safety is the third quality in use characteristic in ISO 9126 - 4. considered measures are listed in following subsections.

User health and safety: What is the incidence of health problems among users of the product?

Safety of people affected by the use of the system: What is the incidence of hazard to people affected by the use of the system?

Economic damage: What is the incidence of economic damage?

Software damage: What is the incidence of software corruption?

5.8.1.4 Satisfaction

Satisfaction is fourth quality in use characteristic in ISO 9126 - 4. Considered measures are listed in following subsections.

Satisfaction scale: How satisfied is the user?

Satisfaction questionnaire: How satisfied is the user with specific software features?

Discretionary usage: What proportion of potential users choose to use the system?

5.8.1.5 Understandability

Users should be able to select a software product, which is suitable for their intended use. An external understandability measure should be able to assess whether new users can understand:

- whether the software is suitable
- how it can be used for particular tasks

Completeness of description What proportion of functions (or types of functions) is understood after reading the product description?

Demonstration accessibility What proportion of the demonstrations/ tutorials can the user access?

Demonstration accessibility in use What proportion of the demonstrations / tutorials can the user access whenever user actually needs to do during an operation?

Demonstration effectiveness What proportion of functions can the user operate successfully after a demonstration or tutorial?

Evident functions What proportion of functions (or types of function) can be identified by the user based on the start-up conditions?

Function understandability What proportion of the product functions will the user be able to understand correctly?

Understandable input and output Can users understand what is required as input data and what is provided as output by software system?

5.8.1.6 Learnability

An external learnability measure should be able to assess how long users take to learn how to use particular functions and the effectiveness of help systems and documentation.

Learnability is strongly related to understandability, and understandability measurements can be indicators of the learnability potential of the software.

Ease of function learning: How long does the user take to learn to use a function?

Ease of learning to perform a task in use: How long does the user take to learn how to perform the specified task efficiently?

Effectiveness of the user documentation and/or help system: What proportion of tasks can be completed correctly after using the user documentation and/or help system?

Effectiveness of the user documentation and/or help systems in use: What proportion of user functions can be used correctly after reading the documentation or using help systems?

Help accessibility: What proportion of the help topics can the user locate?

Help frequency: How frequently does a user have to access help to learn the operation to complete his/her work task?

Operational consistency in use: How consistent are the component of the UI?

5.8.1.7 Operability

An external operability measure should be able to assess whether users can operate and control the software.

- suitability of the software for the task
- self-descriptiveness of the software
- controllability of the software
- conformity of the software with user expectations
- error tolerance of the software

- suitability of the software for individualization

The choice of functions to test will be influenced by the expected frequency of use of functions, the criticality of the functions, and any anticipated usability problems.

Error correction: Can user easily correct error on tasks?

Error correction in use: Can user easily recover his/her error or retry tasks?

Default value availability in use: Can user easily select parameter values for his/her convenient operation?

Message understandability in use: Can user easily understand messages from software system?

Is there any message that caused the user a delay in understanding before starting the next action?

Can user easily memorize important message?

Self- explanatory error messages: In what proportion of error conditions does the user propose the correct recovery action?

Operational error recoverability in use: Can user easily recover his/her worse situation?

Time between human error operations in use: Can user operate the software long enough without human error?

Undo-ability (User error correction): How frequently does the user successfully correct input errors?

Customizability: Can user easily customize operation procedures for his/her convenience?

Can a user, who instructs end users, easily set customized operation procedure templates for preventing their errors?

What proportion of functions can be customized?

Operation procedure reduction: Can user easily reduce operation procedures for his/her convenience?

Physical accessibility: What proportion of functions can be accessed by users with physical handicaps?

5.8.1.8 Attractiveness

An external attractiveness metric should be able to assess the appearance of the software, and will be influenced by factors such as screen design and color.

Attractive interaction: How attractive is the interface to the user?

Interface appearance customizability: What proportion of interface elements can be customized in appearance to the user's satisfaction?

5.8.1.9 Usability compliance

An external usability compliance metric should be able to assess adherence to standards, conventions, style guides or regulations relating to usability.

Usability compliance: How completely does the software adhere to the standards, conventions, style guides or regulations relating to usability?

5.8.2 Literature defined usability measures

Usability measures that are mostly used in literature are listed and defined in following subsections (Nielsen, 1994; Sauro and R.Lewis, 2012; Tullis and Albert, 2013).

5.8.2.1 Task success

Perhaps the most widely used performance measure. It measures how effectively users can complete a given set of tasks. Two different types of task success are binary success and levels of success.

5.8.2.2 Task times

A common performance measure that measures how much time is required to complete a task.

5.8.2.3 Errors

Errors reflect the mistakes made during a task. Errors can be useful in pointing out particularly confusing or misleading parts of an interface.

5.8.2.4 Efficiency

Efficiency can be assessed by examining the amount of effort a user spends to complete a task, such as the number of taps in a mobile application.

5.8.2.5 Learnability

Most applications, especially new ones, require some amount of learning. Usually, learning does not happen in an instant but occurs over time as experience increases. Experience is based on the amount of time spent using an application and the variety of tasks performed.

5.8.2.6 Issues-based measures

Most usability professionals probably consider identifying usability issues and providing design recommendations the most important part of their job. A usability issue might involve confusion around a particular term or piece of content, method of navigation, or just not notic-

ing something that should be noticed. To characterize usability issues we should give some examples as below:

- Anything that prevents task completion
- Anything that creates some level of confusion
- Anything that produces an error
- Not seeing something that should be noticed
- Assuming something is correct when it is not
- Assuming a task is complete when it is not
- Performing the wrong action
- Misinterpreting piece of content
- Not understanding the navigation

5.8.2.7 Self-reported measures

Perhaps the most obvious way to learn about usability of something is to ask users to tell examiners about their experience with it. However, exactly how to ask them so that examiners get adequate data is not so obvious.

CHAPTER 6

iOS APPLICATION USABILITY EVALUATION MODEL

This chapter presents phase 2 of the research methodology (See Figure 4.1). This phase has the following inputs:

- Definitions of usability
- Guidelines to consider during application design
- ISO and literature defined usability measures
- Apple HIG
- Standardized usability questionnaires

This phase provides an iOS application usability evaluation model which will consist of the following:

- Expert-based iOS application usability evaluation criteria
- A user-based iOS application usability evaluation questionnaire
- Apple App Store user rating evaluation

6.1 Expert-based iOS application usability evaluation criteria

To be able to evaluate an iOS application's usability, the definition of evaluation criteria is needed. To define the evaluation criteria, considering different guidelines and platform characteristics is necessary. In this research project a list of usability evaluation criteria for iOS applications has been developed taking the following sources into the consideration:

- ISO defined usability measures

- Literature defined usability measures
- Guidelines to consider during application design
- Apple iOS HIG

Answers are choices from 1 to 5 for each criterion, with choices 4 and 5 considered as an indication of better overall application usability. Criteria are divided into 21 logical groups that are described in more detail in the following sub-sections. Some of the criteria could belong to more than one group. For the sake of simplicity, criteria that could belong to more than one group is listed in the most relevant group.

Table 6.1 presents the list of 79 proposed criteria for expert-based usability evaluation of iOS APPs. These criteria are grouped into 21 (A to V) categories.

Some of the criteria, can be applied to other domains such as usability evaluation of web applications and desktop applications. Criteria which are specific only to iOS applications are marked with * in the table.

Table 6.1 Proposed expert-based usability evaluation criteria

No	Category	Criteria
	A	**Simplicity**
1	1	App uses visual weight and balance to show users the relative importance of onscreen elements. (Pareto Guideline)
2	2	The UI is appropriate for the user's task and skill level and makes it easy to focus on the main task by elevating important content or functionality.
3	3 *	App does not ask user to save when it is not necessary.
4	4 *	Always an obvious and safe way to exit a modal task is provided, to reassure users that their work is safe when they dismiss a modal view.
5	5	The number and prominence of controls is minimized, in order to decrease their weight in the UI. (Hick-Hyman Law)
	B	**User control and navigation**
6	1	The path of navigation is predictable, and markers, such as the back button, are provided to inform users where they are and how to retrace their steps.

Table 6.1 Proposed expert-based usability evaluation criteria (continued)

No	Category	Criteria
7	2	The user can leave an unwanted state via clearly marked cancel/exit points and without having to embark on an extended UI interaction.
8	3 *	The user knows where he is in the app, how he got there, and where he can go via a navigation controller stack and accurate view names.
9	4 *	A back button or gesture returns the app to a previous view without loss of data.
	C	**Understandability**
10	1	The number of controls from which the user must choose is minimized.
11	2	The app's purpose and usage area can be readily understood from the start.
12	3	The user doesn't need to use workarounds or manuals.
13	4	All the views are displayed consistently, so that users can apply knowledge gained in one part of the app to the system as a whole.
14	5 *	The app is consistent with the usage paradigms of builtin APPs, with the same screen navigation hierarchy, content listing style, and mode switching capability using the tab bar.
	D	**Linguistic Clarity**
15	1	Understandable terminology is used in app, that is, words and phrases that are appropriate for the targeted user groups, in all text-based communications.
16	2	Abbreviations and acronyms are not used in an app, unless they are straightforward and easily understood.
	E	**Ease of user input data**
17	1	The requested user input is balanced with what the app offers the user in return, providing as much information or functionality as possible for each piece of information entered by the user.
18	2 *	Making choices is easy for the user, e.g. by providing a table view or a list picker component instead of a text field.
19	3	Required fields are made clear to the user via visual indicators.
20	4	The app validates the information that the user enters into data forms, informing him if it is not in an acceptable format.
21	5 *	Information from the device is obtained when it makes sense to do so, so that users aren't obliged to provide information that is easily accessible by the app.
22	6	The app supports undo and redo.
	F	**Collaboration and Connectedness**
23	1 *	Users are able to easily share information that is important to them, like their location, opinions, and high game scores, when it is appropriate

Table 6.1 Proposed expert-based usability evaluation criteria (continued)

No	Category	Criteria
24	2 *	The app keeps the user informed about the send/receive status of content via a progress indicator.
	G	**Settings**
25	1 *	Settings about preferred app behaviors and information that users rarely want to change in the app included only when it is appropriate to do so.
26	2 *	Users can easily set their preferred behaviors by using the configuration options in the app.
	H	**Branding**
27	1	Brand colors or images presented appropriately in a subtle and understated way for greatest effect.
	I	**Searching**
28	1 *	Local data is live-filtered, so that the app can display results more quickly, narrowing them as the user continues to type.
29	2 *	Remote data is filtered while the user types when possible, informing him that he can opt out if the response time is likely to delay the results by more than a second or two.
30	3 *	Search bars displayed above lists, or lists have index.
31	4 *	Search function featured as a distinct mode if it is a primary function in the app, and search tabs provided only in special circumstances.
32	5 *	Placeholder content and partial results are displayed as they become available to give users prompt access.
33	6 *	Scope bar is provided if the data sort naturally into different categories, as this allows users to specify locations or rules in a search, or to filter objects by specific criteria.
	J	**Application description**
34	1	There is not any spelling, grammatical, and punctuation errors, to avoid creating a negative impression of an app's quality.
35	2	All-capital-letter words kept to a minimum, as they can make text very difficult to read.
	K	**User interface structure**
36	1	Information is conveyed in a condensed, headline-type style, so that users can absorb it quickly and easily.
37	2	The most frequently used (usually higher level) information are placed near the top, and in the following order: from general to specific, and from high level to low level.
38	3	Shortcuts have been developed for the most frequently used parts of the app.

Table 6.1 Proposed expert-based usability evaluation criteria (continued)

No	Category	Criteria
39	4	Labels and titles are consistent throughout the app, and accurately define the tasks to be performed in the app.
40	5 *	Focus on the primary content is maintained in all orientations, so that users feel they have control over the app and the content they care about.
41	6 *	Tappable elements in an app have a target area of about 44 x 44 points, as this size is important for ease of use. (Fitts Law)
42	7	Tappable and untappable areas of the app are clearly recognizable.
43	8 *	App responses to text size changes properly
	L	**User Interface consistency**
44	1	Short labels or well-understood symbols are given to controls, so that users know what they are doing at a glance.
45	2 *	Standard controls and gestures are appropriately and consistently used, so that they behave the way the user expects them to.
46	3 *	The appearance of a controls that perform standard actions is not changed radically , as users will spend time discovering how to use them and wonder what, if anything, this control does that the standard one does not.
47	4	The UI conforms to the user's expectations, in that it meets the predictable contextual needs of the user and respects commonly accepted conventions.
48	5 *	Standard buttons and icons did not used to mean something else.
49	6	UI controls are customized and they are integrated with app's graphical style, and can be discovered and understood without being conspicuous.
50	7	App avoids using the same color in both interactive and non-interactive elements
	M	**Physicality and Realism**
51	1	Similar UI controls are grouped close to each other. Similarity occurs when objects look similar to one another, and can be perceived as part of a group or pattern.
52	2	Related UI controls placed close to each other. Proximity occurs when elements are placed close together, and can be perceived as belonging to a group.
53	3	Figures (forms, silhouettes, and shapes) are differentiated from background (the surrounding area).
	N	**Aesthetic integrity**
54	1	The look of high-quality or precious materials are replicated and materials look realistic and valuable.
55	2	Relevant metaphors representing real-life objects are used when needed to help the user understand, and learn, the task.

Table 6.1 Proposed expert-based usability evaluation criteria (continued)

No	Category	Criteria
56	3	Color coding is used for clarity where appropriate.
57	4	The number of colors is limited to 3-4.
58	5	Beautiful, high-resolution artwork and icons have designed and used in application in accordance with fibonacci sequence.
59	6	Views are designed in compliance with the rule of thirds and golden ratio guidelines hence UI controls are placed in proper positions
60	7 *	App supports retina display
	O	**Subtle Animation**
61	1 *	App makes custom animation consistent with built-in animation when it is appropriate
62	2 *	Animations are used consistently throughout the app, so that users can rely on the experience it gives them.
63	3 *	Uses animation and interactivity to engage users and help them learn by doing
	P	**Gestures**
64	1 *	The actions associated with the standard gestures that users know are not changed.
65	2 *	Complex gestures, or less common ones like swipe or pinch open, are applied as shortcuts to expedite a task, not as the only way to perform a task.
	Q	**Rapidity**
66	1 *	A launch image is displayed which closely resembles the first screen of the app, to decrease the app's perceived launch time.
67	2 *	Displaying an About window or a splash screen is avoided, to ensure that users are not prevented from using the app immediately.
68	3	The login requirement is delayed for as long as possible, to enable users to navigate through much of the app and access some of its functionality without logging in.
69	4 *	App restores its state when restarts, so that users don't have to remember how they had reached it in the first place.
70	5	App avoids asking people to supply setup information
71	6 *	App is fast and responsive to touch events
	R	**Help**
72	1	The app provides easily accessible help to users when needed.
73	2	The Help documentation is properly prepared, and is both appropriate and informative.
74	3	The user can easily move between Help and the current task
75	4	Help is context-based, and addresses all the necessary contexts.
	S	**Error correction and prevention**

Table 6.1 Proposed expert-based usability evaluation criteria (continued)

No	Category	Criteria
76	1 *	Specific bug fixes that customers have been waiting for are specified in the description of a new version of an app.
	T	**In App Purchases and Ads**
77	1 *	App offers in App Purchases and Ads
78	2 *	In app purchases and Ads affect the usability
	V	**Missing Functionalities**
79	1	There are missing features which make app less usable

The following sub-sections explain each criterion in detail.

6.1.1 A - Simplicity

One of the most important aspects of usability is simplicity, and the aim of this group of criteria is to evaluate the simplicity of iOS applications, in other words, to evaluate if application is simple to use. To achieve this goal, five different criteria developed considering Pareto guidelines, Hick-Hyman law, and iOS HIG.

A1 - App uses visual weight and balance to show users the relative importance of on-screen elements. (Pareto Guideline): Giving higher visual weights to more important UI controls/elements and size them with the same importance level the same simplifies the usage of applications. This criterion evaluates if application's UI is designed properly taking into account the visual weights and balance of UI controls and elements.

A2 - The UI is appropriate for the user's task and skill level and makes it easy to focus on the main task by elevating important content or functionality: Each application has a different intended audience. It should be easy for any intended user to focus on the main task because users tend to use their iOS applications for specific purposes and are not going to spend a lot of time on APPs because of device limitations. The UI of applications should be

designed with the skill level of it is audiences in mind. The app should give more priority to more important content or functionality and place them according to their importance.

A3 - App does not ask the user to save when it is not necessary: Built in iOS applications do not ask users to save. Whenever a user changes any setting in any application, application adapts itself to the new change without asking the user to perform an extra step to save. It is recommended to avoid asking user to save if it is not necessary. This criterion evaluates if app asks the user to save even if it is not necessary.

A4 - Always an obvious and safe way to exit a modal task is provided, to reassure users that their work is safe when they dismiss a modal view: Modal tasks are the type of tasks that block user from other interactions in the application. Modal tasks are used in the cases that a mandatory task should be performed without interfering with other parts of the application. Modal tasks help to show the importance of the task and simplify the inference between tasks, but it is necessary to provide an obvious and safe way to exit a modal task. User needs to be assured that the data for other steps or tasks performed by them are safe before exiting a modal task. This criterion evaluates if modal tasks are used properly and provide obvious ways to exit.

A5 - The number and prominence of controls are minimized, in order to decrease their weight in the UI. (Hick-Hyman Law): More UI controls and elements in a UI means more complexity. It is possible to minimize the number and prominence of controls to simplify the UIs. This criterion evaluates the UI to check if there is any unnecessary UI control or is there any way to place them in a different UI to simplify the main UI.

6.1.2 B - User control and navigation

Because of the small size of the smartphones, less information or UI elements fit on the screen to compare with desktop and web applications. Therefore, mobile applications have more screens and a different hierarchy than other types of applications. Having more screens cause

more navigation paths and communication between these screens. This group of criteria evaluates application in terms of user control and navigation over the application UIs.

B1 - The path of navigation is predictable, and markers, such as the back button, are provided to inform users where they are and how to retrace their steps: Applications are usable if the user can easily understand the navigation structure and predict the next steps. This criterion evaluates the navigation structure in the application.

B2 - The user can leave an unwanted state via clearly marked cancel/exit points and without having to embark on an extended UI interaction: Canceling and exiting from unwanted states in the application should be easy for the user. This criterion evaluates if it is easy to leave an unwanted state via clearly marked cancel/exit points.

B3 - The user knows where he is in the app, how he got there, and where he can go via a navigation controller stack and accurate view names: Standard iOS applications use a navigation controller stack to handle the order of navigation. It is convenient for the user to see the path and to know how he got there and where he can go from his current state. This criterion evaluates the application in terms of the navigation clarity for the user.

B4 - A back button or gesture returns the app to a previous view without loss of data: Users do not want to loose the data they have entered into views because of the navigation. Users need to go back and forth between different views without losing data. This criterion evaluates application in terms of data handling during the navigations.

6.1.3 C - Understandability

Users like an application if they can understand app's purpose, goal, and the way it works easily. This group of criteria evaluates the understandability of the applications.

C1 - The number of controls from which the user must choose is minimized: Smaller screen sizes bring less real estate for UI controls. Giving many options/configurations to choose from to the user in the applications makes applications' usage harder and less understandable.

It is better to simplify the options and make it easier for the user to preselect some information without asking the user. This criterion evaluates if application provides minimized number of selections in the application.

C2 - The app's purpose and usage area can be readily understood from the start: This criterion evaluates if application is designed and developed in the way that the user can easily understand the usage area of it from the beginning.

C3 - The user does not need to use workarounds or manuals: If the application is designed with the understandability in mind, user will understand the application's purpose and the way it works without using any extra workarounds or manuals. Some functionalities may need manuals, but the general purpose of application and the way it works should be understandable without using any manuals. This criterion evaluates if application is understandable without use of workarounds or manuals.

C4 - All the views are displayed consistently so that users can apply knowledge gained in one part of the app to the system as a whole: Consistent usage of UI elements and controls, gestures, navigation and actions simplifies the learning process and the usage of the application. User can easily apply the knowledge gained in one part to the other parts if the paradigms are used consistently across the application. This criterion evaluates the consistency of the UI.

C5 - The app is consistent with the usage paradigms of built-in APPs, with the same screen navigation hierarchy, content listing style, and mode switching capability using the tab bar: Built-in applications developed by Apple follow the same screen navigation hierarchy, content listing style, and mode switching capabilities. Users already use the built-in applications and understand the structure hence applications are more easily understandable if they follow the same usage paradigms. This criterion evaluates the consistency of the application with built-in application usage paradigms.

6.1.4 D - Linguistic clarity

The language in the applications is very important for the users to understand the concepts and to be able to use the application in desired extent. Language should be clear and appropriate for the targeted user groups. This group of criteria evaluates the application in terms of linguistic clarity.

D1 - Understandable terminology is used in the app, that is, words and phrases that are appropriate for the targeted user groups, in all text-based communications: Chosen words and phrases in all text-based communications in the application should be understandable by the targeted user groups. Using a terminology that is not understandable by the intended users decreases the understandability of the application and its usability. This criterion evaluates the terminology appropriateness of the application.

D2 - Abbreviations and acronyms are not used in an app unless they are straightforward and easily understood: Having a reference list of the used abbreviations and acronyms is necessary for the text-based communications. User will need to check the meaning of the abbreviations and acronyms from the list but because of small screen sizes in the smart phones this will add an extra navigation step and will complicate the usage. On the other hand, users tend to use iOS applications for quickly completing a task or searching for a concept to quickly grasp. Having unnecessary abbreviations and acronyms will decrease the usability. This criterion evaluates the degree of abbreviations and acronyms usage.

6.1.5 E - Ease of user input data

Some applications provide functionalities that need the input data from the user. It is harder to input data in iOS applications than desktop and web applications because of the device characteristics. This group of criteria evaluates the ease of user input data in the UI.

E1- The requested user input is balanced with what the app offers the user in return, providing as much information or functionality as possible for each piece of information

entered by the user: Entering information about the iOS applications is not as easy as desktop or web applications because of the on-screen keyboard that is harder to use and the size of screens. Therefore, applications should ask as less as data input as possible. In other words, they should ask the data input if it is necessary to provide the proper functionality. This criterion evaluates the balance between the requested user inputs and what app offers the user in return.

E2 - Making choices is easy for the user, e.g. by providing a table view or a list picker component instead of a text field: Because of the on-screen keyboard and smaller screens, it is always easier to make choices via pickers or table views than entering the text. This criterion evaluates if application minimizes the text field inputs and uses easier to use UI controls instead.

E3 - Required fields are made apparent to the user via visual indicators: Required and optional fields should be distinguishable for the user so user can decide if he wants to supply only required information quickly or more to achieve better possible results in his task. This criterion evaluates if required data inputs are marked required.

E4 - The app validates the information that the user enters into data forms, informing him if it is not in an acceptable format: Most of the iOS applications need a backend to communicate for heavy processing of the data. It is going to take less time to validate user data entries than contacting the backend for the data validation. Therefore, it is preferable to handle the data validation in the application client side as soon as possible and notify the user for possible change needs. This criterion evaluates the user data entry validation capabilities of the application.

E5 - Information from the device is obtained when it makes sense to do so, so that users are not obliged to provide information that is easily accessible by the app: Nowadays, mobile devices are computers with extra functionalities in terms of location-awareness, integrated calendar, identification, integrated contacts and built-in functionalities such as camera and maps. Applications can use these functionalities without asking this kind of information to

the user. This criterion evaluates if application obtains the accessible information form device instead of asking the user for the data entry.

E6 - The app supports undo and redo: Undo and redo functionalities are essential in data entry process to correct the mistakes and accelerate the data entry process. This criterion evaluates application to examine if it supports the undo and redo of the data inputs.

6.1.6 F - Collaboration and connectedness

Users like social applications and the capabilities to share and express their idea/feelings about things they care about with other people. iOS applications can improve the collaboration and connectedness of their users with other users by sharing capabilities. This group of criteria evaluates the collaboration and connectedness capabilities of the applications.

F1 - Users can easily share information that is important to them, like their location, opinions, and high game scores, when it is appropriate: Users like to share the content or events with other users via different sharing channels such as email, Facebook, twitter, messages,... This criterion evaluates the application in terms of sharing capabilities.

F2 - The app keeps the user informed about the send/receive status of content via a progress indicator: iOS applications send and receive content from their backends or different sharing services. It is important for the users to know when application communicates with servers via internet usage. Showing progress indicators will notify the users about internet usage and connectivity so user can wait for the process to finish or cancel for unwanted data usages. This criterion evaluates applications in terms of send/receive status feedback.

6.1.7 G - Settings

Users like to be able to change the settings of the applications but giving too many options, and configurations may increase the complexity of the application. This group of criteria evaluates settings capabilities and appropriateness of the applications.

G1 - Settings about preferred app behaviors and information that users rarely want to change in the app included only when it is appropriate to do so: Designers and developers need to think about the severity and necessity of candidate changes in settings when design applications. Adding too many customizations and setting options increases the complexity of the applications and needed to be avoided. This criterion evaluates applications in terms of suitability of settings in the application and checks if there are any unnecessary settings that can be removed.

G2 - Users can easily set their preferred behaviors by using the configuration options in the app: Users like, personalizing and customizing their application's behaviors. This criterion evaluates application regarding the richness and appropriateness of configuration options.

6.1.8 H - Branding

Design of applications should be unique and resemble the brand that application belongs to so user can easily correlate look and feel of the application with the brand he trusts. Users will be able to distinguish their intended brand application from non-official or scam applications. Branding will help the user feel more comfortable because the branded look will be similar to other brand applications such as web and desktop applications. This criterion evaluates if application branded properly.

H1 - Brand colors or images presented appropriately in a subtle and understated way for greatest effect: Branding is important but not more than the overall esthetics and consistency of the design of the application. The usage of brand colors or images should be delicate. This criterion evaluates if brand colors or images are presented appropriately in a subtle and understated way.

6.1.9 I - Searching

Because of the screen size limitations and multiplicity of views it is harder to find the proper content in mobile applications. Proper implementation of searching can be a great help to solve this issue. This group of criteria evaluates the applications in terms of searching capabilities.

I1 - Local data is live-filtered so that the app can display results more quickly, narrowing them as the user continues to type: This criterion evaluates if application live-filters the search results for quickly showing refined results.

I2 - Remote data is filtered during the user data entry when possible, informing him that he can opt out if the response time is likely to delay the results by more than a second or two: This criterion evaluates if application remotely filters the data according to user input informing the user that he can opt out if the filtering is going to take more than a second or two.

I3 - Search bars displayed above lists, or lists have index: This criterion evaluates if application provides search bars above lists or lists have indexes to help the user find the intended content easily.

I4 - Search function featured as a distinct mode if it is a primary function in the app, and search tabs provided only in special circumstances: Search function is a primary function of some of the applications. These applications feature a search function in a distinct mode and provide search placements in special circumstances. If the search function is not the primary function, application can handle this function differently. This criterion evaluates if search function is the primary function and how it is handled.

I5 - Placeholder content and partial results are displayed as they become available to give users prompt access: This criterion evaluates the way application presents partial results of the search.

I6 - Scope bar is provided if the data sort naturally into different categories, as this allows users to specify locations or rules in a search, or to filter objects by specific criteria: The

scope bar allows the user to refine his search by limiting the search results to a specific scope or category. This criterion evaluates if application provides scope bar in searches.

6.1.10 J - Application description

This group of criteria evaluates application regarding the application description in the App Store and application itself.

J1 - There is not any spelling, grammatical, and punctuation errors, to avoid creating a negative impression of an app's quality: Spelling, grammatical, and punctual errors in the application and App Store descriptions give users a negative impression of app's quality. This criterion evaluates application regarding the spelling, grammatical, and punctual errors.

J2 - All-capital-letter words kept to a minimum, as they can make text very difficult to read: All-capital letter words make the text in the application and App Store application description very difficult to read. This criterion evaluates if all-capital-letter words are used properly.

6.1.11 K - User interface structure

This group of criteria evaluates the UI structure of the application.

K1 - Information is conveyed in a condensed, headline-type style so that users can absorb it quickly and easily: Users can absorb information quickly and easily if the given information is structured properly in the application. This criterion evaluates if application presents the information in a condensed, headline-type style.

K2 - The most frequently used (usually higher level) information are placed near the top, and are ordered from general to specific, and from high level to low level: Users understand the content quickly and easily if the information is presented top-down (from general to specific, and from high level to low level) consistently. This criterion evaluates if application presents the information in the proper structure.

K3 - Shortcuts have been developed for the most frequently used parts of the app: Users use some functionalities or content of the application more frequently than other parts. Providing shortcuts for the most frequently used parts of the application helps users to use application quickly and easily. This criterion evaluates if application provides necessary shortcuts to the user.

K4 - Labels and titles are consistent throughout the app, and accurately define the tasks to be performed in the app: It is easier to understand and learn the application if the labels and titles accurately define the criterion to be performed and are consistent throughout the application. This criteria evaluates the naming convention consistency of the application.

K5 - Focus on the primary content is maintained in all orientations so that users feel they have control over the app and the content they care about: Users feel they have control over the application and its content if the structure and focus on the primary content is maintained in all orientations. This criterion evaluates the way application handles orientations.

K6 - Tappable elements in an app have a target area of about 44 * 44 points, as this size is important for ease of use (Fitts Law): Users easily interact with the application if tappable UI elements are proper for their finger size. It is recommended that tappable elements in an app have 44 * 44 points (Average user finger size). This criterion evaluates if tappable elements in the application are in the proper size to make it easy for tapping.

K7 - Tappable and untappable areas of the app are clearly recognizable: Easily recognizing the tappable areas of an application makes it easier for the user to use the application. This criterion evaluates if it is easy to recognize the tappable and untappable areas of the application.

K8 - App responses to text size changes well: iOS 7 provides a setting to change the size of the text globally. Applications can develop this functionality to change their text size when user changes the text size in the setting application. This criterion evaluates if application responses to text size change appropriately and maintains the information structure.

6.1.12 L - User interface consistency

This group of criteria evaluates the UI consistency of the application.

L1 - Short labels or well-understood symbols are given to controls, so that users know what they are doing at a glance: Naming the UI controls with consistent short labels or well-understood symbols simplifies the understandability of the application. This criterion evaluates if application gives the proper labels and symbols to the UI controls.

L2 - Standard controls and gestures are appropriately and consistently used so that they behave the way the user expects them to: Users expect standard UI controls and gestures behave like how they behave in the built-in iOS applications because they got used to it that way. This criterion evaluates if standard controls and gestures are appropriately and consistently used in the application.

L3 - The appearance of controls that perform standard actions is not changed radically, as users will spend time discovering how to use them and wonder what, if anything, this control does that the standard one does not: Users spend time discovering how to use the controls that perform standard actions if their appearance is changed radically. This criterion evaluates the degree application changes the appearance of UI controls that perform standard actions.

L4 - The UI conforms to the user's expectations, in that it meets the predictable contextual needs of the user and respects commonly accepted conventions: This criterion evaluates if the application UIs conform to the user expectations by respecting commonly accepted conventions and predictable contextual needs.

L5 - Standard buttons and icons did not use to mean something else: This criterion evaluates if application uses standard buttons and icons properly.

L6 - UI controls are customized, and they are integrated with app's graphical style, and can be discovered and understood without being conspicuous: Customizing the UI controls

and integrating them into the application's graphical style make application more appealing but users use the UI controls easily if the UI controls can be discovered and understood quickly. This criterion evaluates if the application customizes and integrates the UI controls with application's graphical style considering the discoverability and understandability of the UI control.

L7 - App avoids using the same color in both interactive and non-interactive elements: Using different colors for interactive and non-interactive UI elements helps users to differentiate intractable and un-intractable areas of the application. This criterion evaluates if application uses different colors to emphasize intractability.

6.1.13 M - Physicality and realism

This group of criteria evaluates application UI design in terms of realism and physicality.

M1 - Similar UI controls are grouped close to each other. Similarity occurs when objects look similar to one another and can be perceived as part of a group or pattern: This criterion evaluates if similar UI controls are grouped close to each other.

M2 - Related UI controls placed close to each other. Proximity occurs when elements are placed close together, and can be perceived as belonging to a group: This criterion evaluates if related UI controls are placed close to each other.

M3 - Figures (Forms, silhouettes, and shapes) are differentiated from background (the surrounding area): This criterion evaluates if figures such as forms, silhouettes and shapes are differentiated from their surrounding area.

6.1.14 N - Aesthetic integrity

This group of criteria evaluates the esthetics and its integrity with the application context.

N1 - The look of high-quality or precious materials are replicated, and materials look realistic and valuable: Replicating the look of high-quality or precious materials and using

them in the application can improve the esthetics of the application but these materials should look realistic and valuable to make application esthetically appealing. This criterion evaluates application if replicated materials look realistic and valuable.

N2 - Relevant metaphors representing real-life objects are used when needed to help the user understand and learn, the task: Metaphors can help the user understand and learn, the task if they are relevant and represent real-life objects. Using metaphors can improve the esthetics of an application though. This criterion evaluates if application uses relevant metaphors to improve the understandability and learnability.

N3 - Color coding is used for clarity where appropriate: Using compatible colors representing different types of content and/or UI controls can improve the understandability and esthetics of the application. This criterion evaluates if application uses proper color coding.

N4 - The number of colors is limited to 3-4: Limiting the number of colors used in the application to 3-4 colors can simplify the UI and make it more appealing. This criterion evaluates if application limits the number of colors to 3-4.

N5 - Beautiful, high-resolution artwork and icons have designed and used in the application in accordance with Fibonacci sequence: Having their own beautiful, high-resolution artworks and icons makes applications visually appealing. This criteria evaluates if application has a beautiful high-resolution artwork and icons integrated into its design.

N6 - Views are designed in compliance with the rule of thirds and golden ratio guidelines hence UI controls are placed in proper positions: Placing the UI elements in proper positions on the screen can improve the esthetics of the applications. Applications can use the rule of thirds and golden ratio guidelines to place elements in the proper positions. This criterion evaluates regarding the UI positioning.

N7 - App supports Retina display: Retina displays on iOS devices have higher resolution and sharper view. Applications can support these displays by providing properly sized artwork, image, and icons. This criterion evaluates if application supports Retina display.

6.1.15 O - Subtle animation

This group of criteria evaluates the way application uses animations.

O1 - App makes custom animation consistent with built-in animation when it is appropriate: Animations can help the interactivity and user engagement. Applications can have custom animations in the iOS application. Consistency of the custom animations with built-in animations helps the understandability and learnability of the application though. This criterion evaluates if application custom animations are consistent with built-in animations.

O2 - Animations are used consistently throughout the app, so that users can rely on the experience it gives them: Using animations consistently throughout the application improves the user experience. This criterion evaluates if animation usage is consistent and understandable throughout the application.

O3 - App uses animation and interactivity to engage users and help them learn by doing: This criterion evaluates the in application on-boarding experience and the degree of animation usage in the application to help the user learning how to use application.

6.1.16 P - Gestures

This group of criteria evaluates application regarding gesture usage.

P1 - The actions associated with the standard gestures that users know are not changed: Built-in iOS applications use standard gestures. Applications can have custom gestures as well as standard gestures. Users are familiar with standard actions for standard gestures therefore changing the behavior of these gestures in application may lead to problems in terms of learnability and understandability. This criterion evaluates applications if they associate standard actions with standard gestures.

P2 - Complex gestures or less common ones like swipe or pinch open, are applied as shortcuts to expedite a task, not as the only way to perform a task: Some of the gestures

are complex or are not used much in built-in iOS applications, so regular users are not familiar with them. It is recommended to apply these gestures as shortcuts to expedite a task not, as the only way to perform a task. This criterion evaluates if applications use complex and uncommon gestures properly.

6.1.17 Q - Rapidity

This group of criteria evaluates application regarding its quickness and rapidity.

Q1 - A launch image is displayed which closely resembles the first screen of the app, to decrease the app's perceived launch time: Launch of iOS applications may take some time depending on the necessary setup to start the application. Displaying a proper launch image makes application look faster. However, application launch should not take much time and prevent a user using application rapidly. This criteria evaluates the usage of launch image.

Q2 - Displaying an About window or a splash screen is avoided, to ensure that users are not prevented from using the app immediately: Applications should make the user able to use application as soon as possible. To ensure that users are not prevented from using the application immediately, displaying views such as about or splash screen is not recommended. This criterion evaluates if application presents an about or a splash screen and prevents the user from immediate usage of the application.

Q3 - The login requirement is delayed for as long as possible, to enable users to navigate through much of the app and access some of its functionality without logging in: Some of the iOS applications need users to log in to their credentials to be able to use them. Logging in is not necessary to be able to use all the application functionalities in most cases though. Users tend to install the application and explore it in their iOS device. Asking for the login without showing any other functionalities block the users that want to explore and understand before registering. Therefore, it is recommended to provide some functionalities that do not need log into the user so user can understand the application better before the registration. This criterion evaluates if application provides some functionality without login.

Q4 - App restores its state when restarts, so that users do not have to remember how they had reached it in the first place: This criterion evaluates if application preserves usage states and restores its state when restarts so users can continue to their task right away.

Q5 - App avoids asking people to supply setup information: Some applications may need to ask users some questions for the setup. Asking many questions for setup prevents users from using application as soon as possible. It is recommended to minimize the setup process and use device data such as location, camera, contact list,... as much as possible. This criterion evaluates the setup process of the application.

Q6 - App is fast and responsive to touch events: Any iOS application supports a specific version of iOS and above. Applications should work fast and be responsive to touch events for specific versions of the iOS that they support. This criterion evaluates if application runs smoothly on supported versions of iOS.

6.1.18 R - Help

This group of criteria evaluates application in terms of help functionalities.

R1 - The app provides easily accessible help to users when needed: This criterion evaluates the help functionality and accessibility of help when it is necessary.

R2 - The Help documentation is properly prepared and is both appropriate and informative: This criterion evaluates the quality of help documentation.

R3 - The user can easily move from Help to the current task: This criterion evaluates if user can use help with his current task without having problem in going back and forth.

R4 - Help is context-based, and addresses all the necessary contexts: Some applications provide only general help, and some provide context-based specific help. It is recommended to provide the context-based help though. This criterion evaluates if the help is context-based and addresses all the necessary contexts.

6.1.19 S - Error correction and prevention

This group of criteria evaluates error correction and prevention of the application. This study assumes that applications are tested properly before publishing them to the App Store and they are not buggy.

S1 - Specific bug fixes that customers have been waiting for are specified in the description of a new version of an app: It is recommended to fix bugs and improve the quality of the application frequently. Informing the users about these changes in the App Store helps the user experience though. This criteria evaluates the application in terms of bug fixing and informing the user about them.

6.1.20 T - In-App purchases and Ads

This group of criteria evaluates application in terms of in-app purchases and ads in the application and the degree of which they affect the usability.

T1 - App offers in App Purchases and Ads: Some applications do not offer all the functionalities directly and sell those functionalities via in-app purchases or ads. Some applications offer ads in the application as well. This criterion evaluates if application offers in-app purchases and provides ads in the application.

T2 - In-app purchases and ads affect the usability: In-app purchases and ads can affect the usability of the application. It is not possible to use some of the applications because the user needs to buy those functionalities. Some ads also decrease the user experience quality. This criterion evaluates if in-app purchases and ads affect the usability badly.

6.1.21 V - Missing functionalities

This group of criteria evaluates applications in terms of the necessary functionality.

V1 - There are missing features that make an app less usable: Absence of some functionalities makes application less usable. This criterion evaluates if there are missing features that make application less usable.

6.2 User-based iOS application usability evaluation criteria

This research study has examined different standardized user-based evaluation questionnaires. Ryu (2005) has been used as the major reference to construct the user-based evaluation questionnaire. Ryu (2005) is tailored for mobile phones and has 124 questions but is not up to date since it does not consider new mobile application characteristics. This research study proposes a post-study questionnaire with 20 application specific questions, and only aims for end users.

Table 6.2 presents the resulting questionnaire for user-based APP usability evaluation.

6.3 Apple App Store user rating evaluation

This section evaluates Apple's method to calculate the App Store APP user ratings and proposes an improved method to classify the user ratings as positive and negative classes.

Apple App Store is the only source to buy and download applications for the iOS. There are over a million of applications in the App Store. Each application in the App Store has a page that describes the application and gives information about its functionality, version changes, category, development company, languages that it supports, compatible iOS versions and devices, in-app purchases, customer reviews, and ratings.

Users tend to check the application page in the App Store to examine screenshots or videos related to the application. Reading customer reviews and examining user given ratings give much information about the application before buying or downloading any application.

Users give 1 to 5 stars to the application to express the degree of satisfaction with the application. Higher numbers express better user ratings. Apple App Store provides the count of given stars for each category (one star, two stars,..., five stars) and presents an average rating by cal-

Table 6.2 Proposed user-based usability evaluation criteria

No	Criteria
1	Is it easy to learn how to use this application?
2	Is it easy to navigate between hierarchical menus and pages?
3	Is the presentation of system information sufficiently clear and understandable?
4	Are the command names meaningful?
5	Are the input and text entry methods for this application easy and usable?
6	Is application easily customizable for your needs?
7	Is the design of the graphic symbols, icons and labels on the icons sufficiently relevant?
8	Is it easy to search and find relevant information in the application?
9	Is the organization of the menus and information on the application screen sufficiently clear and logical?
10	Is the application interface sufficiently similar to those of other applications you have used?
11	Is this application attractive and pleasing?
12	Are the response time and information display fast enough?
13	Is the HELP information given by application useful?
14	Is it easy to take corrective actions once an error has been recognized?
15	Has the application at some time stopped unexpectedly?
16	Does application have all the functions and capabilities you expect it to have?
17	Is using this application frustrating?
18	Would you miss this application if you no longer had it?
19	Are you/would you be proud of this application?
20	Is the application reliable, dependable, and trustworthy?

culating the mean of the ratings considering their weight. The user ratings are ordinal numbers that the distance between numbers are not defined hence it is not appropriate to calculate the mean of these numbers.

This research study calculates the overall user rating of an application in the following manner:

$$userRatingRatio = \frac{numberOfFours + numberOfFives}{numberOfOnes + numberOfTwos} * 100 \tag{6.1}$$

This equation gives the ratio of 4 and five stars over one and two stars. One can assume that the given three stars correspond broadly to mid-range rating, so this research study omits them in the calculation.

If the calculated ratio is equal or over 50, then the user rating is considered as positive; otherwise it is considered as negative:

$$userRating = \begin{cases} Positive, & \text{if } userRatingRatio >= 50 \\ Negative, & \text{otherwise} \end{cases} \tag{6.2}$$

CHAPTER 7

EXPERIMENTS ON iOS APPLICATION USABILITY EVALUATION

This chapter presents phase 3 of the research methodology (See Figure 4.1). This phase has the following inputs:

- Expert-based iOS application usability evaluation criteria
- Apple App Store user rating evaluation
- List of selected applications from the Apple App Store

This phase provides the following outputs:

- Expert-based evaluation results for 99 APPs
- Apple App Store user rating classes (positive|negative) for 99 APPs
- Dataset for the experiments in phase 5
- Dataset analysis

7.1 Evaluation process

Machine learning models need a historical dataset that can be used for learning, about the problem that they try to solve. At the beginning of this study, there was not any suitable dataset available from the literature. A proper dataset could have information about the usability of the applications along with their Apple App Store user ratings.

Therefore, in the course of this study, a usability evaluation model for iOS applications has to be developed (see chapter 6).

The proposed evaluation process has seven steps as follows:

- Selecting the APPs from the Apple App Store
- Testing the applicability of the proposed expert-based evaluation criteria with a subset of 11 iOS APPs
- Evaluating and peer-reviewing 39 APPs with the all proposed expert-based evaluation criteria by three researchers
- Updating the expert-based evaluation criteria according to the feedbacks of researchers
- Evaluating and peer-reviewing 60 APPs with the updated expert-based evaluation criteria
- Re-evaluating the 39 APPs with the updated criteria
- Constructing the dataset by combining the evaluation results of 99 APPs with their Apple App Store user rating classes

The following sub-sections explain each step in detail.

7.1.1 Selection of APPs from the Apple App Store

One of the researchers randomly selected 99 different APPs considering the following selection criteria:

- Application should be rated at least by 100 users
- Application should belong to one of the following categories
  - Productivity
  - Utility
  - Business
  - Health
  - Finance

  - Life style
  - Weather
  - References
  - Food & Beverages
  - Music
  - Sports
  - Photos
- Application should be complex enough to be evaluated in terms of usability

40.4% of applications have the negative user ratings and 59.6% of applications have the positive user ratings. Appendix IV gives the list of these applications including their types and user ratings. The same researcher downloaded/purchased the selected applications and installed them on two iPhone 4 devices running iOS 7+.

7.1.2 Testing the applicability of the proposed expert-based evaluation criteria with a subset of 11 iOS APPs

Devices and applications were ready for the experiments, but researchers wanted to test the evaluation criteria on a small subset of selected applications. Therefore, two of the researchers used a subset of criteria to evaluate 11 APPs, and the applicability of the proposed criteria for the selected applications has been documented in Nayebi *et al.* (2013).

7.1.3 Evaluating and peer-reviewing 39 APPs with all proposed expert-based evaluation criteria by three researchers

Three researchers evaluated 39 of the selected APPs including the 11 previously evaluated APPs. Each researcher evaluated the applications alone and, in the end, the three researchers peer-reviewed the results to agree on the final results. This process was very time-consuming, but it was necessary to ensure the reliability of the results.

7.1.4 Updating the expert-based evaluation criteria according to the feedbacks of researchers

During the evaluation phase, the researchers have faced problems regarding the applicability and testability of the some of the criteria. For instance, it was very hard to test the error correction and prevention related criteria. Also, researchers commented that a number of criteria could be added to improve the evaluation model. Criteria related to in-app purchases and ads and missing functionalities are some of the examples. Therefore, the researchers updated the expert-based evaluation criteria regarding the feedbacks.

7.1.5 Evaluating and peer-reviewing 60 APPs with the updated expert-based evaluation criteria

The researchers evaluated 60 additional APPs with the revised evaluation criteria. Evaluators peer- reviewed the evaluation results and finalized the results for 60 applications.

7.1.6 Re-evaluating 39 APPs with the updated criteria

The next step was to re-evaluate, and peer-review the previously evaluated 39 applications using the revised criteria. At the end of this step, the dataset contained the results of the evaluation of each criterion for 99 applications: in other words, all 79 independent variables for all 99 instances.

7.1.7 Constructing the dataset by combining the evaluation results of 99 APPs with their Apple App Store user rating classes

A complete dataset for the machine learning model should have the dependent variable for each instance. In this study, each instance is an evaluated APP, with 79 independent variables and APP's Apple App Store user rating is the dependent variable.

Next step was to add the user rating for each application to the dataset. The Apple App Store user rating class of each APP calculated according to the definition in the section 6.3. The researchers added the resulted user rating classes to each instance to construct the dataset.

At the end of the APP usability evaluation experiment, the dataset has 99 instances representing 99 applications with 79 independent variables (in this study, values assigned to criteria by the researchers) and the dependent variable (positive/negative user rating class value).

7.2 Descriptive statistics and Dataset analysis results

This section presents the descriptive statistics definitions and the dataset analysis results.

7.2.1 Central tendency of criteria

Mean (average value), median (middle value) and mode (most frequent value) are measures of central tendency, which give us information about the middle or center of a distribution. In this study, mean is not used because the data has ordinal values.

The following subsections present the central tendency related descriptive statistics used in this research study.

7.2.1.1 Median

The median is the middle value in a set of ordered data values. It is the value that separates the higher half of a dataset, a population, or a probability distribution, from the lower half. The median of a dataset of N values for a criteria x, sorted in increasing order is the middle value. If there is an odd number of observations, then the median is the middle value of the ordered dataset. If N is even, then there is no single middle value; it is the mean of the two middle values.

$$median = l_1 + (\frac{N/2 - (\sum freq)_l}{freq_{median}})width \tag{7.1}$$

where l_1 is the lower boundary of the median interval, N is the number of values in the entire dataset, $(\sum freq)_l$ is the sum of the frequencies of all of the intervals that are lower than the median interval, $freq_{median}$ is the frequency of the median interval, and $width$ is the width of the median interval.

Figure 7.1 presents the distribution of medians. The X-axis presents the median value and Y-axis presents the count of APPs that have the corresponding median value. For instance, 20 of APPs have the median value of 5.

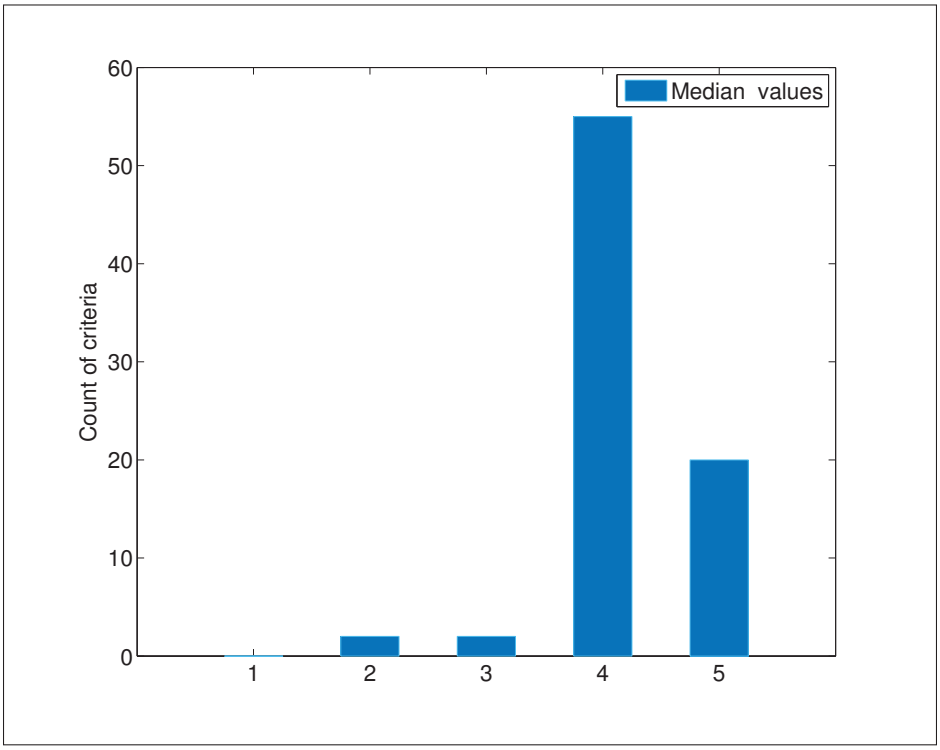

Figure 7.1 Dataset - Distribution of Median values for the criteria

The distribution of median values presented in the Figure 7.1 can be summarized as the following:

- The median is four for 55 of 79 criteria; in other words, 60% of criteria have the median of four. This means that researchers gave many fours to many criteria.

- 20 of the criteria have a median of 5.
- None of the criteria have the median of one while there are two criteria with median of two and two with a median of three.

7.2.1.2 Mode

Mode is the value that appears most frequently in a dataset. Mode is a way of expressing, in a single number, necessary information about a random variable or a population. The numerical value of mode is the same as that of the mean and median in a normal distribution, and it may be very different in highly skewed distributions.

Figure 7.2 presents the distribution of modes.

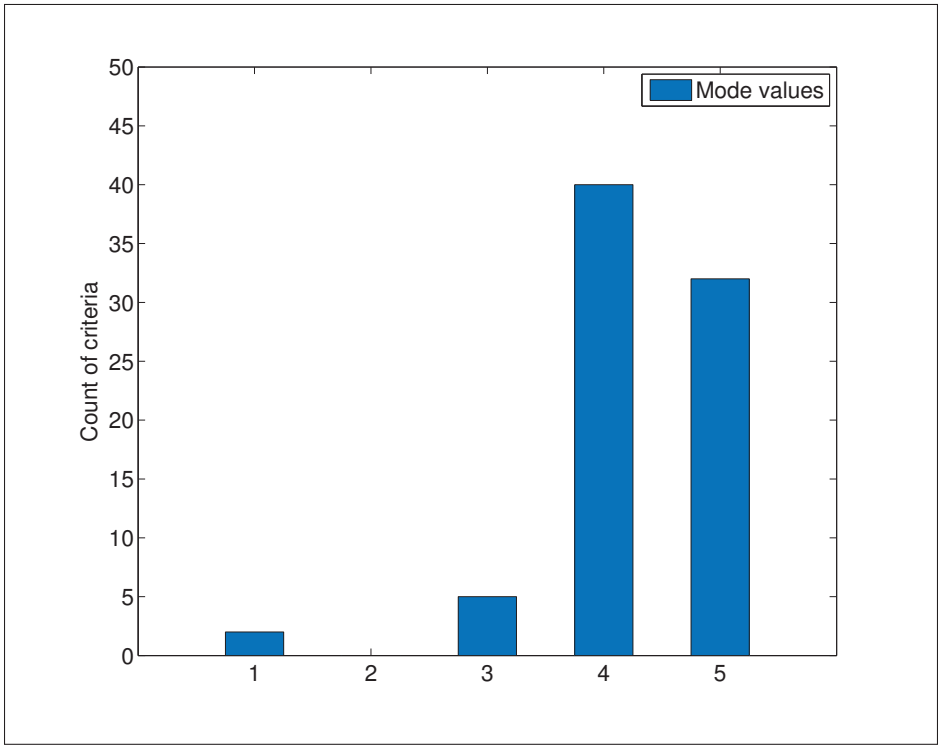

Figure 7.2 Dataset - Distribution of Mode values for the criteria

The distribution of mode values presented in the Figure 7.2 can be summarized as the following:

- 40 of 79 criteria have the mode of four hence the researchers gave four more than other numbers to the criteria.
- 32 of 79 criteria have the mode of five.
- Only two criteria have the mode of one: this means that the evaluators gave mostly one to two of criteria.
- There are not any criteria with the mode of two.
- There are four criteria with a mode of three.

7.2.2 Dispersion of criteria

Dispersion (also called variability, scatter, or spread) denotes how stretched or squeezed a distribution (theoretical or that underlying a statistical sample) is. Typical examples of measures of statistical dispersion are the variance, standard deviation, and interquartile range.

7.2.2.1 Variance

The variance is a measure of variability. It measures how far a set of numbers is spread out. A variance of zero indicates that all the values are identical. Variance is always non-negative: a small variance indicates that the data points tend to be very close to the mean (also called the expected value) and hence to each other, while a high variance indicates that the data points are very spread out around the mean and from each other. Variance is the sum of the squared distances of dataset value from the mean divided by the variance divisor. The variance of N observations (in this study, evaluation results) $x_1, x_2, ..., x_N$, for a numeric criteria X is

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{n}(x_i - \bar{x})^2 = (\frac{1}{N}\sum_{i=1}^{n}x_i^2) - \bar{x}^2 \tag{7.2}$$

where $\bar{x}$ is the mean value of the evaluation results.

Figure 7.3 presents the Variance values for each criterion in a scatter chart.

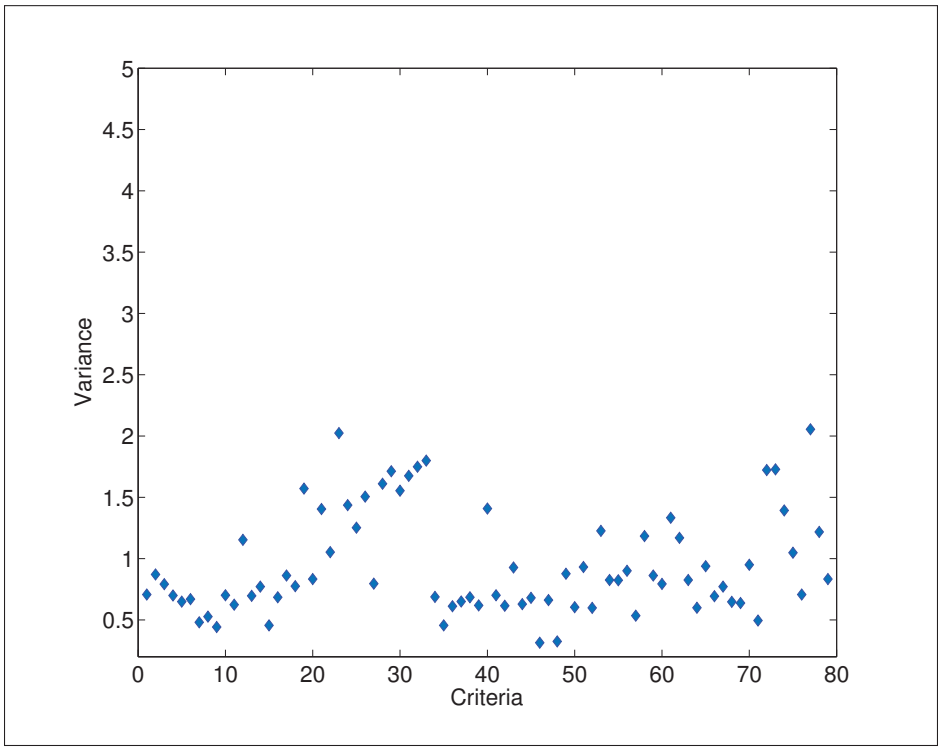

Figure 7.3 Dataset - Variance values for each criterion

The distribution of variance values presented in the Figure 7.3 can be summarized as the following:

- Variance value is not zero, but it is low for most of the criteria.
- Variance values change between 0.324 and 2.055 for the criteria.

Presented lower variance values mean that the values of a criterion for all 99 evaluated applications are close to the mean and each other. In other words, researchers gave mostly fours and fives to the criteria.

7.2.2.2 Standard Deviation

Standard deviation is a measure that is used to quantify the amount of variation or dispersion of a dataset values. Standard deviation is the square root of the variance. A low standard deviation indicates that the observations (in this study, evaluation results) tend to be very close to the mean (also called the expected value) of the dataset, while a high standard deviation means that the observations are spread out over a wider range of values.

Figure 7.4 presents the Standard Deviation values for each criterion in a scatter chart.

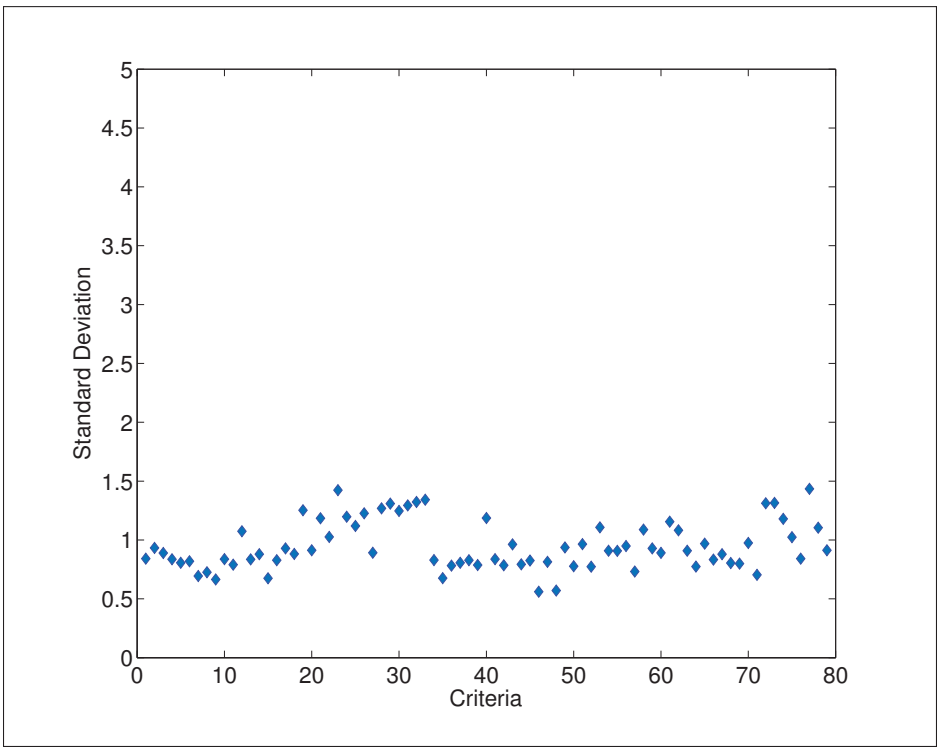

Figure 7.4 Dataset - Standard Deviation values for each criterion

The distribution of standard deviation values presented in the Figure 7.4 can be summarized as the following:

- Standard deviation value is not zero, but it is low for the most of the criteria.

- Standard deviation values change between 0.570 and 1.423 for the criteria.

Standard deviation is low for the criteria. Therefore, values of the evaluated criteria for all 99 applications are close to the mean and each other. In other words, researchers gave mostly fours and fives to the criteria.

7.2.2.3 Standard Error of Mean

The standard error of the mean (SEM) is the standard deviation of the sample mean's estimate of a population mean. It can also be viewed as the standard deviation of the error in the sample mean with respect to the actual mean since the sample mean is an unbiased estimator. SEM is usually estimated by the sample estimate of the population standard deviation (sample standard deviation) divided by the square root of the sample size (assuming statistical independence of the values in the sample):

$$SE_x = \frac{s}{\sqrt{n}} \tag{7.3}$$

where

- s is the sample standard deviation (i.e., the sample-based estimate of the standard deviation of the population), and
- n is the size (number of observations) of the sample.

This estimate may be compared with the formula for the true standard deviation of the sample mean:

$$SD_x = \frac{\sigma}{\sqrt{n}} \tag{7.4}$$

where σ is the standard deviation of the population.

Figure 7.5 presents the SEM values for each criterion in a scatter chart.

The distribution of standard error of mean values presented in the Figure 7.5 can be summarized as the following:

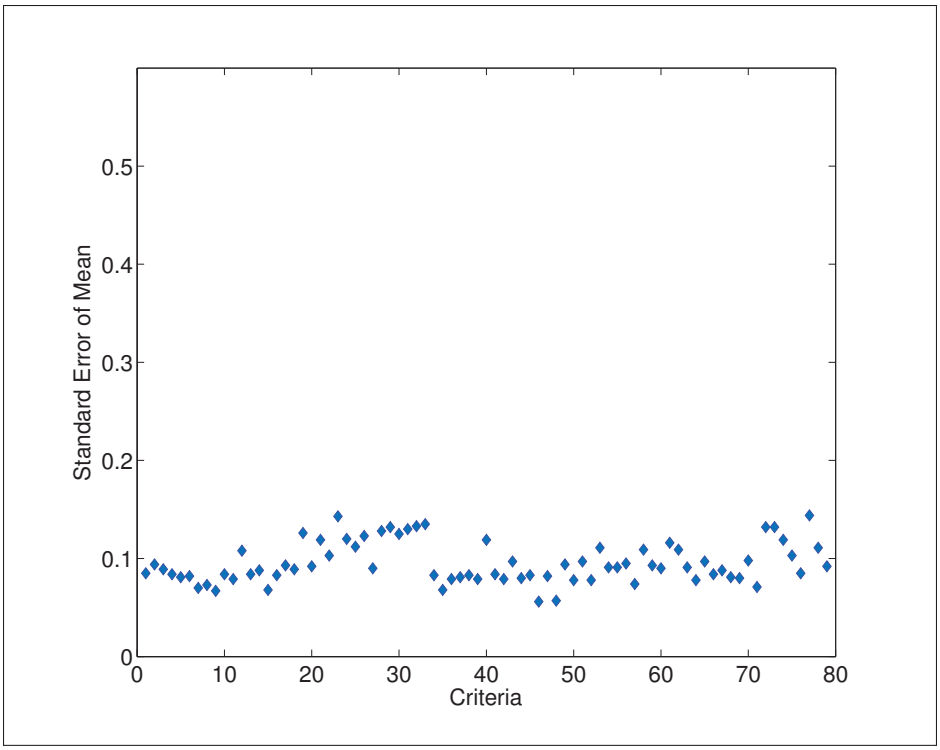

Figure 7.5 Dataset - Standard Error of Mean values for each criterion

- Standard error of the mean value is not zero, but it is low for the most of the criteria.
- Standard error of mean values change between 0.056 and 0.144 for the criteria.

The SEM for each criterion is slightly smaller than its standard deviation and is low for the criteria. Therefore, values of the evaluated criteria for all 99 applications are close to the mean and each other.

7.2.2.4 Skewness

Skewness is a measure of the asymmetry of the probability distribution of a real-valued random variable about its mean. The skewness value can be positive or negative, or even undefined. Skewness measures the degree and direction of asymmetry. A symmetric distribution such as a normal distribution has a skewness of 0, and a distribution that is skewed to the left, e.g. when the mean is less than the median, has a negative skewness.

Figure 7.6 presents the Skewness values for each criterion in a scatter chart.

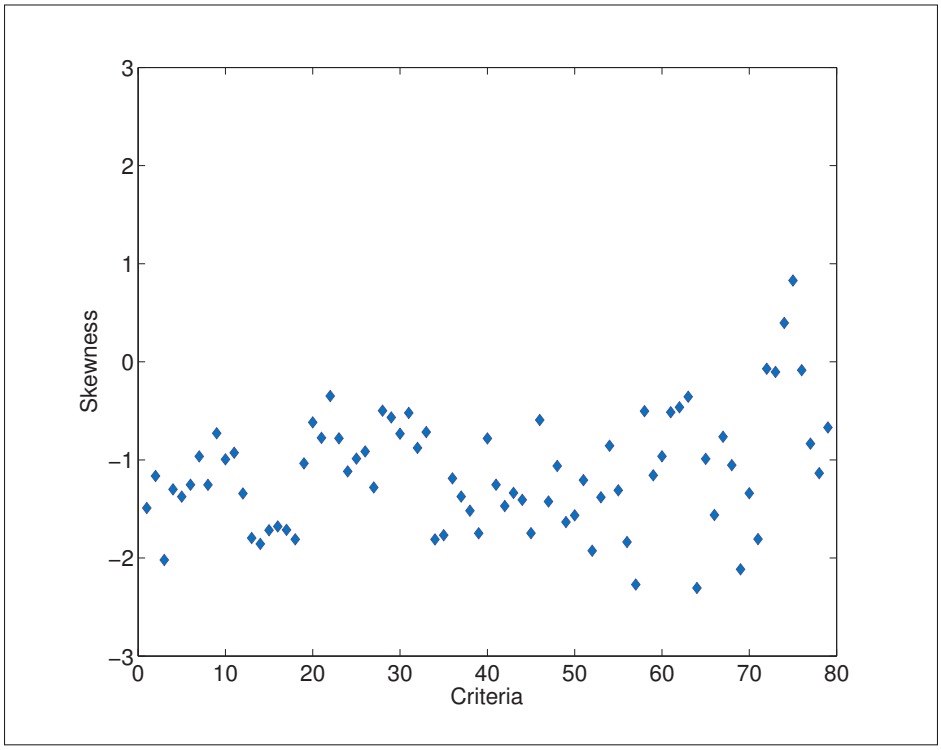

Figure 7.6 Dataset - Skewness values for each criterion

The distribution of skewness values presented in the Figure 7.6 can be summarized as the following:

- Skewness values for the majority of the criteria are negative.
- Skewness values change between -2.306 and 0.829.
- Most of the criteria have the negative skewness values.

Most of the criteria have the negative skewness values. Therefore, criteria evaluation results in the dataset are not symmetrical.

7.2.2.5 Kurtosis

Kurtosis quantifies whether the shape of the data distribution matches the Gaussian distribution. In fact, Kurtosis is a measure of the heaviness of the tails of a distribution. In a similar way to the concept of skewness, kurtosis is a descriptor of the shape of a probability distribution and, just as for skewness, there are different ways of quantifying it for a theoretical distribution and corresponding ways of estimating it from a sample from a population. There are various interpretations of kurtosis, and of how particular measures should be interpreted; these are primarily peakedness (width of peak), tail weight, and lack of shoulders (distribution primarily peak and tails, not in between).

Extremely non-normal distributions may have high positive or negative kurtosis values while nearly normal distributions will have kurtosis values close to 0. Kurtosis is positive if the tails are "heavier" than a normal distribution and negative if the tails are "lighter" than a normal distribution.

Figure 7.7 presents the Kurtosis values for each criterion in a scatter chart.

The distribution of kurtosis values presented in the Figure 7.7 can be summarized as the following:

- The majority of the criteria does not have kurtosis value of 0.
- Kurtosis values change between -1.227 and 7.445.

The distribution of most of the criteria does not match the Gaussian distribution because they do not have kurtosis of 0 and are more peaked than a Gaussian distribution.

7.2.2.6 Range

The range of the dataset is the difference between the largest ($max()$) and smallest ($min()$) values. The range provides an indication of statistical dispersion. It is measured in the same units

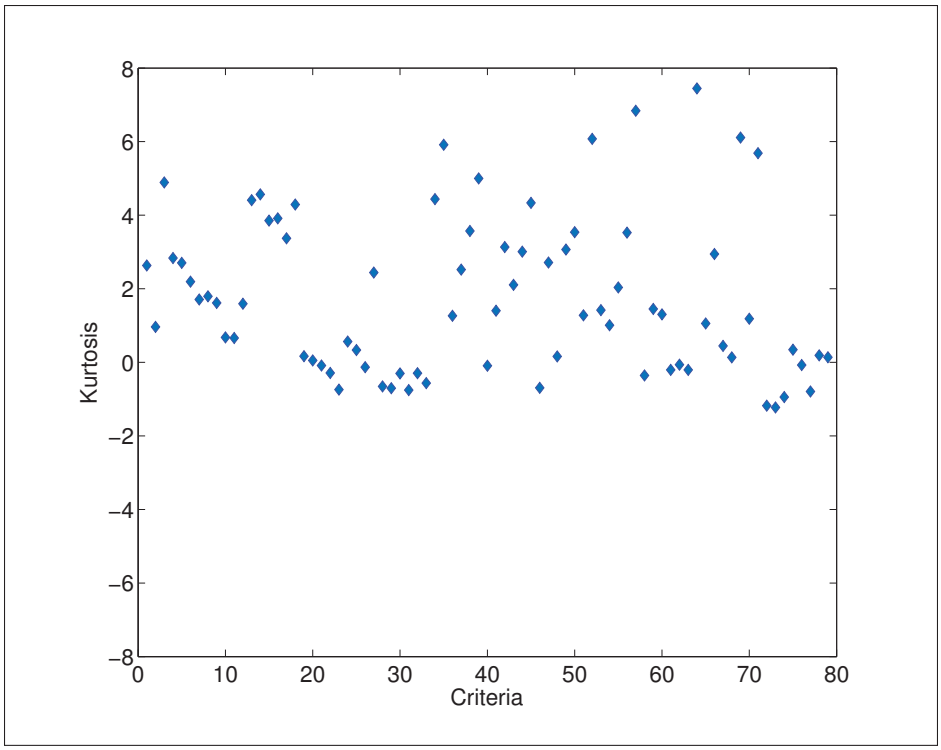

Figure 7.7 Dataset - Kurtosis values for each criterion

as the data. Since it only depends on two of the observations, it is most useful in representing the dispersion of small datasets.

Figure 7.8 presents the Range values for each criterion in a bar chart. Some of the criteria have different range values, but the range is 4 for most of the criteria, so it means that researchers gave ones and fives to the most of the criteria.

7.2.2.7 Percentiles

A percentile (or a centile) is a measure used in statistics indicating the value below which a given percentage of observations in a group of observations falls. For example, the 20th percentile is the value (or score) below which 20 percent of the observations may be found.

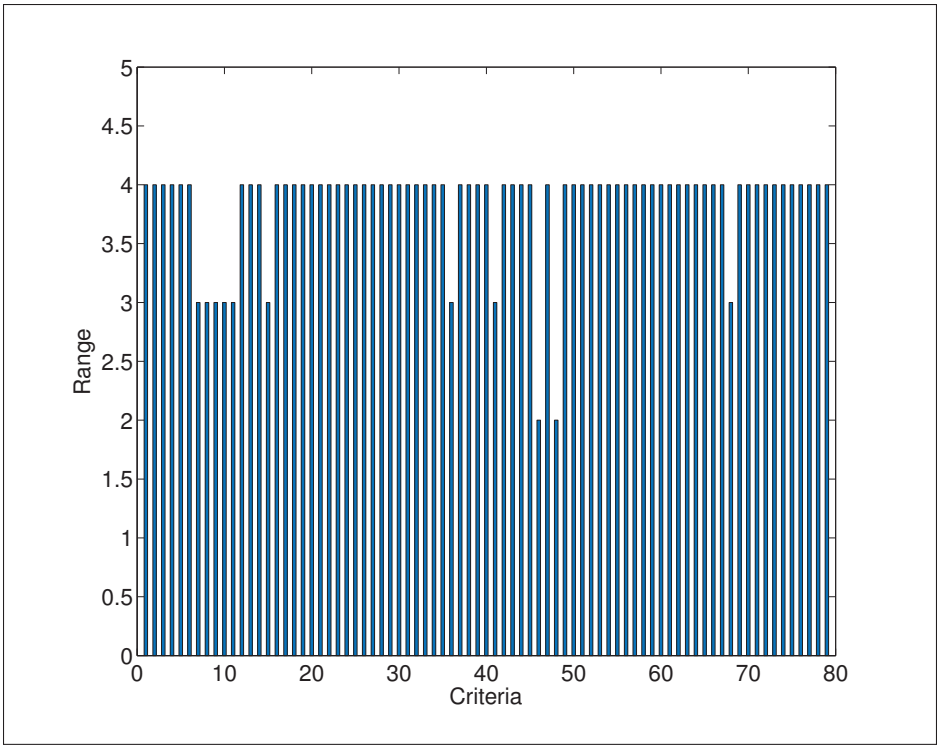

Figure 7.8 Dataset - Range values for each criterion

In this study, percentiles are 25, 50 and 75 and Table 7.1 presents the values for those percentiles. For instance, criteria R2, R3 and R4 have the 1 for percentile 25, 2 or 3 for percentile 50 and 3 or 4 for percentile 75.

7.3 Summary of the dataset analysis

Previous sections have presented the different statistics to understand the dataset and have shown that the dataset has many fours and fives. None of the criteria has the variance of zero. While some of the criteria have variance around 2.00, some have the lower variance values.

Also, dataset is skewed to the left for the most of criteria. This skewness could lead to the wrong inferences. Machine learning models can learn the behavior of dataset including the evaluators' judgments and adapt themselves. Therefore dataset skewness and low variance may not be a problem.

Table 7.1 presents the statistics for each criteria.

Table 7.1 Dataset statistics for each criterion

Criteria	Std. Err of Mean	Median	Mode	Std. Deviation	Variance	Skewness	Kurtosis	Range	Percentiles		
A1	.085	4	5	.841	.707	-1.491	2.633	4	4	4	5
A2	.094	4	5	.933	.871	-1.164	.964	4	4	4	5
A3	.089	5	5	.890	.792	-2.021	4.890	4	4	5	5
A4	.084	4	4	.836	.699	-1.301	2.835	4	4	4	5
A5	.081	4	4	.806	.649	-1.375	2.705	4	4	4	5
B1	.082	4	4	.819	.670	-1.253	2.194	4	4	4	5
B2	.070	4	4	.694	.481	-.965	1.707	3	4	4	5
B3	.073	5	5	.726	.527	-1.255	1.796	3	4	5	5
B4	.067	4	4	.665	.442	-.728	1.615	3	4	4	4
C1	.084	4	4	.837	.701	-.995	.679	3	4	4	5
C2	.079	4	4	.790	.624	-.927	.661	3	4	4	5
C3	.108	4	4	1.074	1.153	-1.342	1.594	4	4	4	5
C4	.084	4	5	.835	.697	-1.796	4.410	4	4	4	5
C5	.088	4	4	.879	.772	-1.857	4.566	4	4	4	5
D1	.068	5	5	.675	.456	-1.717	3.853	3	4	5	5
D2	.083	5	5	.828	.685	-1.679	3.915	4	4	5	5
E1	.093	5	5	.929	.862	-1.714	3.371	4	4	5	5
E2	.089	5	5	.881	.776	-1.810	4.290	4	4	5	5
E3	.126	4	4	1.253	1.571	-1.036	.167	4	3	4	5
E4	.092	4	4	.913	.833	-.618	.052	4	3	4	5
E5	.119	4	5	1.185	1.405	-.775	-.086	4	3	4	5
E6	.103	4	3	1.026	1.053	-.349	-.289	4	3	4	4
F1	.143	4	5	1.423	2.024	-.780	-.742	4	3	4	5
F2	.120	4	4	1.198	1.436	-1.118	.566	4	3	4	5
G1	.112	4	4	1.119	1.252	-.987	.334	4	3	4	5
G2	.123	4	4	1.227	1.506	-.914	-.132	4	3	4	5
H1	.090	4	4	.892	.796	-1.281	2.439	4	4	4	5
I1	.128	4	4	1.269	1.611	-.499	-.656	4	3	4	4
I2	.132	4	4	1.309	1.713	-.567	-.704	4	3	4	4
I3	.125	4	4	1.247	1.554	-.733	-.301	4	3	4	5
I4	.130	4	4	1.294	1.675	-.521	-.753	4	3	4	4
I5	.133	4	4	1.323	1.749	-.878	-.297	4	3	4	5
I6	.135	4	4	1.342	1.800	-.716	-.565	4	3	4	5
J1	.083	5	5	.829	.688	-1.811	4.439	4	4	5	5
J2	.068	5	5	.676	.456	-1.768	5.913	4	4	5	5
K1	.079	4	5	.782	.612	-1.188	1.266	3	4	4	5
K2	.081	4	5	.806	.650	-1.374	2.519	4	4	4	5
K3	.083	4	4	.828	.685	-1.518	3.571	4	4	4	5
K4	.079	4	5	.787	.619	-1.748	4.998	4	4	4	5

Table 7.1 Dataset statistics for each criterion (continued)

Criteria	Std. Err of Mean	Median	Mode	Std. Deviation	Variance	Skewness	Kurtosis	Range	Percentiles		
K5	.119	4	4	1.187	1.408	-.781	-.092	4	3	4	5
K6	.084	4	4	.837	.701	-1.254	1.403	3	4	4	5
K7	.079	4	5	.785	.616	-1.470	3.135	4	4	4	5
K8	.097	4	4	.963	.928	-1.337	2.107	4	4	4	5
L1	.080	4	4	.793	.629	-1.408	3.008	4	4	4	5
L2	.083	5	5	.825	.680	-1.747	4.334	4	4	5	5
L3	.056	5	5	.560	.314	-.594	-.694	2	4	5	5
L4	.082	4	5	.814	.662	-1.425	2.713	4	4	4	5
L5	.057	5	5	.570	.325	-1.062	.161	2	4	5	5
L6	.094	4	4	.937	.877	-1.634	3.067	4	4	4	5
L7	.078	5	5	.777	.604	-1.565	3.538	4	4	5	5
M1	.097	4	4	.965	.932	-1.206	1.277	4	4	4	5
M2	.078	4	4	.773	.598	-1.926	6.075	4	4	4	5
M3	.111	4	4	1.107	1.226	-1.381	1.418	4	4	4	5
N1	.091	4	4	.909	.826	-.856	1.009	4	3	4	4
N2	.091	4	4	.908	.824	-1.310	2.035	4	4	4	5
N3	.095	5	5	.949	.901	-1.838	3.525	4	4	5	5
N4	.074	5	5	.732	.535	-2.271	6.838	4	4	5	5
N5	.109	4	4	1.088	1.184	-.503	-.356	4	3	4	4
N6	.093	4	4	.929	.862	-1.157	1.452	4	4	4	5
N7	.090	4	4	.891	.794	-.963	1.305	4	4	4	5
O1	.116	4	3	1.155	1.333	-.514	-.206	4	3	4	4
O2	.109	4	3	1.082	1.170	-.463	-.063	4	3	4	4
O3	.091	4	4	.909	.826	-.357	-.206	4	3	4	4
P1	.078	5	5	.774	.599	-2.306	7.445	4	4	5	5
P2	.097	4	4	.969	.938	-.990	1.055	4	3	4	5
Q1	.084	4	5	.833	.694	-1.562	2.945	4	4	4	5
Q2	.088	4	4	.879	.772	-.765	.446	4	4	4	5
Q3	.081	5	5	.804	.647	-1.053	.134	3	4	5	5
Q4	.080	5	5	.799	.638	-2.116	6.110	4	4	5	5
Q5	.098	4	5	.975	.950	-1.340	1.185	4	4	4	5
Q6	.071	5	5	.704	.495	-1.807	5.685	4	4	5	5
R1	.132	3	3a	1.312	1.722	-.071	-1.179	4	2	3	4
R2	.132	3	4	1.315	1.728	-.104	-1.227	4	1	3	4
R3	.119	2	1	1.180	1.393	.396	-.943	4	1	2	3
R4	.103	2	1	1.024	1.048	.829	.342	4	1	2	3
S1	.085	4	3	.841	.707	-.085	-.074	4	3	4	4
T1	.144	5	5	1.434	2.055	-.834	-.793	4	3	5	5
T2	.111	5	5	1.104	1.218	-1.135	.189	4	3	5	5
V1	.092	4	4	.913	.833	-.669	.138	4	3	4	4

CHAPTER 8

APPLE APP STORE iOS APPLICATION USER RATING PREDICTION

This chapter presents phase 5 of the research methodology (See Figure 4.1). This phase tests the hypotheses presented in the research objectives:

- H1a - Usability of an iOS application and user given ratings in the Apple App Store are related.
- H1b - It is possible to predict an iOS application's user given ratings by evaluating its usability and constructing a prediction model.

To test the hypotheses this phase experiments different machine learning models with the following inputs:

- Dataset that is explained in the chapter 7.
- Machine learning classification models for the Apple App Store iOS APP user rating prediction that is explained in chapter 4 and depicted in Figure 4.2.

This phase provides the following outputs:

- Experiments
- Performance evaluation results of five machine learning classification models
- Selection of best model for the dataset
- Threats to validity

Chapter 7 has presented the dataset and the statistics. Analysis of the dataset and statistics has concluded that the dataset has many fours and fives and is skewed to the left. This chapter

presents the experiments with the five machine learning classification models on the dataset. Experiments will be evaluated using the indicators presented in the chapter 4. The results of indicator evaluation will present if the dataset is sufficient for predicting Apple App Store user ratings. If preliminary results in the experiments present a low prediction accuracy, a data preprocessing strategy and feature subset selection techniques can be utilized to minimize the consequences of low variance and the skewness.

8.1 Experiments

This study experiments five different machine learning classification models and presents the evaluation of each. The steps of experiments are explained in research methodology (chapter 4) and are depicted in figure 4.2.

Applying the feature subset selection on the dataset decreased the performance of the different models. Therefore, feature subset selection results are not presented in this experimentation. This study experiments with the full dataset that has 79 features and 99 instances. The dataset was used to select the testing and training datasets. This experiment has divided the data into 75% training (74 instances) and 25% (25 instances) testing datasets.

The training dataset is used to train the following models, and the testing dataset is used for the testing:

- Naïve Bayes
- Bayesian Network
- MLP (Multi-Layer Perceptron)
- RBF (Radial Basis Function)
- SVM (Support Vector Machines)

The results of experiments are analyzed and presented in the following Tables and Figures for each of the models:

Prediction results in Figure:

To make the Figure readable and understandable, this study converts the class labels (Positive user rating: TRUE and Negative user rating: FALSE) to zeros and ones and adds a random noise (Jitter) to the zeros and ones.

Predictions in test dataset Table:

This Table gives the Actual vs. Predicted results and marks the error instances

Evaluation on test dataset Table:

This table presents the accuracy and Kappa statistic.

Detailed accuracy by class Table:

This Table presents the indicators such as TP (True positives), FP (False positives), Precision, Recall, and F-Measure.

Confusion Matrix:

This Table presents the confusion matrix of the machine learning classification model.

Furthermore, the models are compared, and the best model is selected for the dataset.

8.1.1 Naïve Bayes

This study applied Naïve Bayes classification model on the dataset to predict the user ratings in Apple App Store. Full dataset used, and dataset split to 75% for training dataset and 25% for testing dataset, therefore, there are 74 instances for the training and 25 instances for the testing.

While 53% of applications have FALSE (Negative user rating) class, 47% have TRUE (Positive user rating) class. Results are presented in the following sub-sections.

8.1.1.1 Predictions on test dataset

Figure 8.1 presents the actuals against the predictions. In this Figure jittered ones represent TRUE and jittered zeros represent FALSE classes. Also, X-axis presents the predicted class, and Y-axis presents the actual class. Four instances that are misclassified have the X (Predicted class) of zero and Y (Actual class) of one. Remaining instances are placed correctly in the intersection of X and Y axes for zeros and ones.

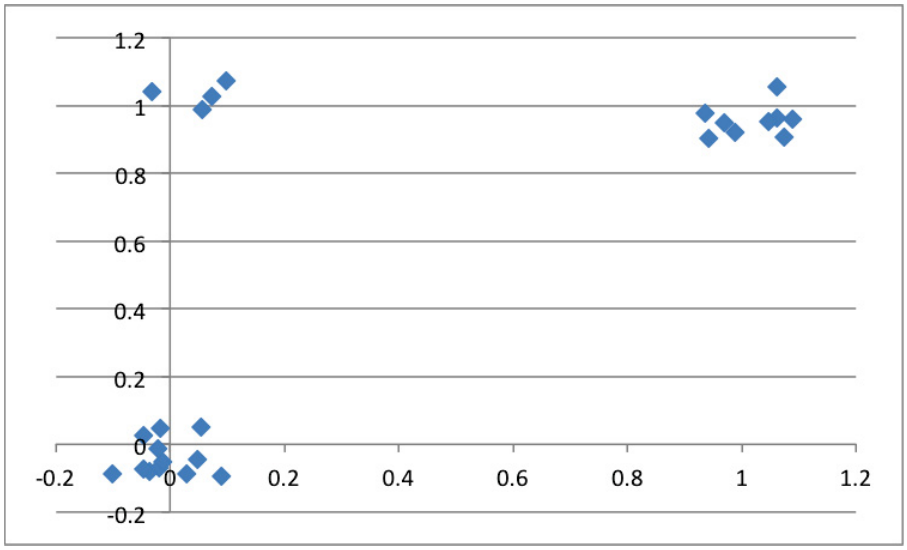

Figure 8.1 Naïve Bayes user rating prediction results - 25 instances

In addition, Table 8.1 presents the predictions on the test dataset.

Misclassified instances are marked with + in error column, and the probability distribution columns present the probability of FALSE class and the probability of TRUE class. For instance, the Naïve Bayes classification model gives the probability of one that instance number 1 belongs to the FALSE class. Higher probabilities of FALSE and TRUE for each instance are marked with * in the table.

Table 8.1 Naïve Bayes prediction results on test dataset

No	Actual	Predicted	Error	Probability Distribution	
				FALSE	TRUE
1	1:FALSE	1:FALSE		*1	0
2	1:FALSE	1:FALSE		*0.991	0.009
3	1:FALSE	1:FALSE		*1	0
4	2:TRUE	2:TRUE		0.193	*0.807
5	1:FALSE	1:FALSE		*1	0
6	2:TRUE	2:TRUE		0.026	*0.974
7	1:FALSE	1:FALSE		*1	0
8	1:FALSE	1:FALSE		*1	0
9	2:TRUE	2:TRUE		0	*1
10	2:TRUE	1:FALSE	+	*1	0
11	1:FALSE	1:FALSE		*1	0
12	2:TRUE	2:TRUE		0.274	*0.726
13	1:FALSE	1:FALSE		*1	0
14	2:TRUE	2:TRUE		0	*1
15	1:FALSE	1:FALSE		*1	0
16	2:TRUE	1:FALSE	+	*0.998	0.002
17	1:FALSE	1:FALSE		*1	0
18	1:FALSE	1:FALSE		*1	0
19	2:TRUE	2:TRUE		0.116	*0.884
20	2:TRUE	2:TRUE		0.035	*0.965
21	2:TRUE	2:TRUE		0.002	*0.998
22	2:TRUE	1:FALSE	+	*1	0
23	2:TRUE	1:FALSE	+	*1	0
24	1:FALSE	1:FALSE		*1	0
25	2:TRUE	2:TRUE		0.127	*0.873

8.1.1.2 Evaluation on test dataset

Table 8.2 presents the results of the evaluation on the testing dataset.

Table 8.2 Naïve Bayes evaluation on test dataset

Correctly Classified Instances	21	84%
Incorrectly Classified Instances	4	16%
Kappa statistic	0.6835	
Total Number of Instances	25	

Naïve Bayes correctly classifies the user rating of 84% of 25 applications in the testing dataset. In other words, four applications are misclassified.

Kappa statistic value is 0.6835 which means that the model agrees to the true classes well.

8.1.1.3 Detailed accuracy by class

The TP for the FALSE class is 1 and is 0.692 for the TRUE class (The greater is, the better). The weighted average TP for the FALSE and TRUE classes is 0.84. The FP for the FALSE class is 0.308 and 0 for the TRUE class (The less is, the better). The WAVG (weighted average) FP for the FALSE and TRUE classes is 0.148.

Precision of the model for the TRUE is one and 0.75 for the FALSE. This means that the model exactly predicts the TRUE but fails in 25% of the FALSE cases to predict exactly.

Recall of the model for the TRUE is 0.692 and one for the FALSE. This means that all the FALSE instances are predicted correctly, and some of the TRUE instances are predicted as FALSE.

Table 8.3 presents the results of model accuracy for each class.

Table 8.3 Naïve Bayes detailed accuracy by class

	TP	**FP**	**Precision**	**Recall**	**F-Measure**	**ROC Area**	**Class**
	1	0.308	0.75	1	0.857	0.878	FALSE
	0.692	0	1	0.692	0.818	0.878	TRUE
WAVG.	0.84	0.148	0.88	0.84	0.837	0.878	

8.1.1.4 Confusion matrix

Table 8.4 presents the confusion matrix for the Naïve Bayes model. According to the confusion matrix, this model classifies all FALSE classes as FALSE and four of the TRUE classes as FALSE also.

Table 8.4 Naïve Bayes confusion matrix

a	b	<– Classified
12	0	| a = FALSE
4	9	| b = TRUE

8.1.2 Bayesian Network

The dataset is split to 75% for training and 25% for testing. Therefore, there are 74 instances (In this study, applications) for the training and 25 instances for the testing. Bayesian Network model is used to predict the Apple App Store user rating, and results are given in the following sub-sections.

8.1.2.1 Predictions on test dataset

Figure 8.2 presents the actuals against the predictions. Two instances that are misclassified have the X (Predicted class) of zero and Y (Actual class) of one. Remaining instances are placed correctly in the intersection of X and Y axes for zeros and ones.

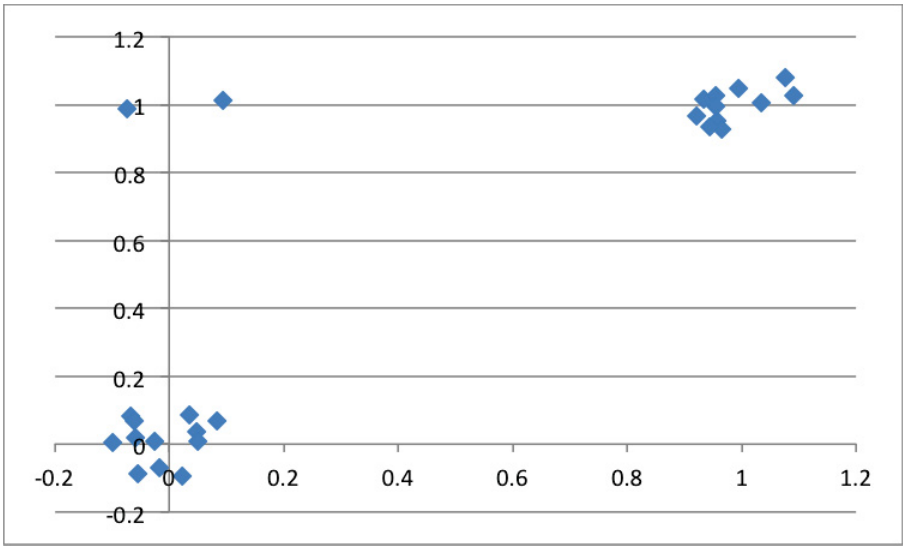

Figure 8.2 Bayesian Network user rating prediction results - 25 instances

In addition, Table 8.5 presents the predictions on test dataset. Only two applications (Application No 10 and 23) are misclassified and are marked with + in error column. They were TRUE, and the model has predicted them as FALSE. The Bayesian Network classification model predicts that these two applications should have negative user ratings, so it is better to improve their usability before publishing them to the App Store. The probability distribution columns present the probability of class FALSE and the probability of TRUE class in order from left to right. For instance, model gives the probability of 0.824 that instance number 10 belongs to the FALSE class but in fact it belongs to the TRUE class. Higher probabilities of FALSE and TRUE for each instance are marked with * in the table.

8.1.2.2 Evaluation on test dataset

Table 8.6 presents the Bayesian Network evaluation results on the test dataset. This model correctly classifies the user rating of 23 of 25 applications in the test dataset, that is: 92% of applications are classified correctly.

The Kappa statistic value is 0.8408 which means there is an excellent agreement between the model and the actual class.

8.1.2.3 Detailed accuracy by class

The TP for the FALSE class is 1 and is 0.846 for the TRUE class (The greater is, the better). The weighted average TP for the FALSE and TRUE classes is 0.92. The FP for the FALSE class is 0.154 and 0 for the TRUE class (The less is, the better). The weighted average FP for the FALSE and TRUE classes is 0.074.

Precision of the model for the TRUE is one and 0.857 for the FALSE. This means that the model exactly predicts the TRUE but fails in 15% of FALSE cases to predict exactly.

Recall of the model for the TRUE is 0.846 and one for the FALSE. This means that all FALSE instances are predicted correctly, and some of the TRUE instances are predicted as FALSE.

Table 8.5 Bayesian Network predictions on test dataset

No	Actual	Predicted	Error	Probability Distribution	
				FALSE	TRUE
1	1:FALSE	1:FALSE		*0.824	0.176
2	1:FALSE	1:FALSE		*0.824	0.176
3	1:FALSE	1:FALSE		*0.824	0.176
4	2:TRUE	2:TRUE		0.411	*0.589
5	1:FALSE	1:FALSE		*0.824	0.176
6	2:TRUE	2:TRUE		0.411	*0.589
7	1:FALSE	1:FALSE		*0.824	0.176
8	1:FALSE	1:FALSE		*0.824	0.176
9	2:TRUE	2:TRUE		0.411	*0.589
10	2:TRUE	1:FALSE	+	*0.824	0.176
11	1:FALSE	1:FALSE		*0.824	0.176
12	2:TRUE	2:TRUE		0.063	*0.937
13	1:FALSE	1:FALSE		*0.824	0.176
14	2:TRUE	2:TRUE		0.411	*0.589
15	1:FALSE	1:FALSE		*0.824	0.176
16	2:TRUE	2:TRUE		0.411	*0.589
17	1:FALSE	1:FALSE		*0.824	0.176
18	1:FALSE	1:FALSE		*0.824	0.176
19	2:TRUE	2:TRUE		0.411	*0.589
20	2:TRUE	2:TRUE		0.063	*0.937
21	2:TRUE	2:TRUE		0.063	*0.937
22	2:TRUE	2:TRUE		0.411	*0.589
23	2:TRUE	1:FALSE	+	*0.824	0.176
24	1:FALSE	1:FALSE		*0.824	0.176
25	2:TRUE	2:TRUE		0.411	*0.589

Table 8.6 Bayesian Network evaluation on test dataset

Correctly Classified Instances	23	92%
Incorrectly Classified Instances	2	8%
Kappa statistic	0.8408	
Total Number of Instances	25	

Table 8.7 presents the Bayesian Network detailed accuracy by class values.

Table 8.7 Bayesian Network detailed accuracy by class

	TP	FP	Precision	Recall	F-Measure	ROC Area	Class
	1	0.154	0.857	1	0.923	0.923	FALSE
	0.846	0	1	0.846	0.917	0.923	TRUE
WAVG.	0.92	0.074	0.931	0.92	0.92	0.923	

8.1.2.4 Confusion matrix

Table 8.8 presents the confusion matrix of the prediction model. Only two of the TRUE cases are classified as FALSE. All FALSE cases are classified correctly.

Table 8.8 Bayesian Network confusion matrix

a	b	<– Classified
12	0 |	a = FALSE
2	11 |	b = TRUE

8.1.3 MLP (Multi-Layer Perceptron)

This research study used different parameters that are explained in research methodology of this thesis and achieved the best results by the 1000 epochs, 0.75 learning rate, 41 hidden layers and 0 seed.

8.1.3.1 Predictions on test dataset

Figure 8.3 presents the actuals against the predictions. Four instances that are misclassified have the X (Predicted class) of zero and Y (Actual class) of one. One instance that is misclassified has the X of one and Y of zero. Remaining instances are placed correctly in the intersection of X and Y axes for zeros and ones.

Table 8.9 presents the MLP prediction results for each of the 25 test instances.

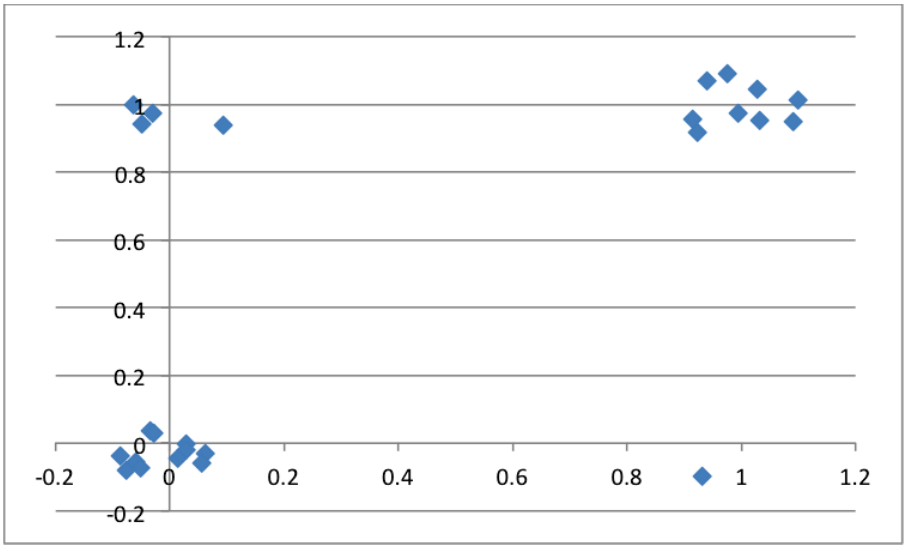

Figure 8.3 MLP (Multi-Layer Perceptron) user rating prediction results - 25 instances

For instance, model gives the probability of 0.868 that instance number 1 belongs to the FALSE class. Higher probabilities of FALSE and TRUE for each instance are marked with * in the table.

8.1.3.2 Evaluation on test dataset

This MLP classification model correctly classifies the user rating of 80% of 25 applications in the testing dataset. Five applications are misclassified.

The Kappa statistic value for this model is 0.6032 which means that there is a good agreement between the model and the actual class.

Table 8.10 presents the MLP evaluation on test dataset.

8.1.3.3 Detailed accuracy by class

The TP for the FALSE class is 0.917 and is 0.692 for the TRUE class (The greater it is, the better). The weighted average TP for the FALSE and TRUE classes is 0.8. The FP for the

Table 8.9 MLP (Multi-Layer Perceptron) predictions on test dataset

No	Actual	Predicted	Error	Probability Distribution	
				FALSE	TRUE
1	1:FALSE	1:FALSE		*0.868	0.132
2	1:FALSE	1:FALSE		*1	0
3	1:FALSE	1:FALSE		*1	0
4	2:TRUE	2:TRUE		0	*1
5	1:FALSE	1:FALSE		*0.999	0.001
6	2:TRUE	2:TRUE		0.006	*0.994
7	1:FALSE	1:FALSE		*0.768	0.232
8	1:FALSE	1:FALSE		*0.998	0.002
9	2:TRUE	1:FALSE	+	*0.784	0.216
10	2:TRUE	1:FALSE	+	*0.64	0.36
11	1:FALSE	1:FALSE		*0.999	0.001
12	2:TRUE	2:TRUE		0.007	*0.993
13	1:FALSE	1:FALSE		*0.843	0.157
14	2:TRUE	2:TRUE		0	*1
15	1:FALSE	1:FALSE		*0.825	0.175
16	2:TRUE	1:FALSE	+	*0.993	0.007
17	1:FALSE	1:FALSE		*0.87	0.13
18	1:FALSE	2:TRUE	+	0.106	*0.894
19	2:TRUE	2:TRUE		0.316	*0.684
20	2:TRUE	2:TRUE		0	*1
21	2:TRUE	2:TRUE		0	*1
22	2:TRUE	2:TRUE		0.172	*0.828
23	2:TRUE	1:FALSE	+	*1	0
24	1:FALSE	1:FALSE		*0.968	0.032
25	2:TRUE	2:TRUE		0	*1

Table 8.10 MLP (Multi-Layer Perceptron) evaluation on test dataset

Correctly Classified Instances	20	80%
Incorrectly Classified Instances	5	20%
Kappa statistic	0.6032	
Total Number of Instances	25	

FALSE class is 0.308 and 0.083 for the TRUE class (The less it is, the better). The weighted average FP for the FALSE and TRUE classes is 0.191.

Precision of the model for the TRUE is 0.9 and 0.733 for the FALSE. This means that the model predicts the 90% of TRUE cases exactly and 73.3% of FALSE cases exactly.

Recall of the model for the TRUE is 0.692 and 0.917 for the FALSE. This means that 91.7% of the FALSE instances are predicted correctly, and 30.8% of TRUE instances are predicted as FALSE.

Table 8.11 presents the MLP detailed accuracy by class.

Table 8.11 MLP (Multi-Layer Perceptron) detailed accuracy by class

	TP	**FP**	**Precision**	**Recall**	**F-Measure**	**ROC Area**	**Class**
	0.917	0.308	0.733	0.917	0.815	0.853	FALSE
	0.692	0.083	0.9	0.692	0.783	0.853	TRUE
WAVG.	0.8	0.191	0.82	0.8	0.798	0.853	

8.1.3.4 Confusion matrix

Table 8.12 presents the MLP confusion matrix. According to the confusion matrix, one of the FALSE instances is predicted as TRUE and four of the FALSE instances are predicted as FALSE.

Table 8.12 MLP (Multi-Layer Perceptron) confusion matrix

a	**b**	**<– Classified**
11	1 |	a = FALSE
4	9 |	b = TRUE

8.1.4 RBF (Radial Basis Function)

The dataset is split to 75% for training and 25% for testing. Therefore, there are 74 applications for the training and 25 applications for the testing. RBF is used to predict the Apple App Store user rating, and results are given in the following sub-sections.

8.1.4.1 Predictions on test dataset

Figure 8.4 presents the Actual against Prediction. Four instances that are misclassified have the X (Predicted class) of zero and Y (Actual class) of one. One instance that is misclassified has the X of one and Y of zero. Remaining instances are placed correctly in the intersection of X and Y axes for zeros and ones.

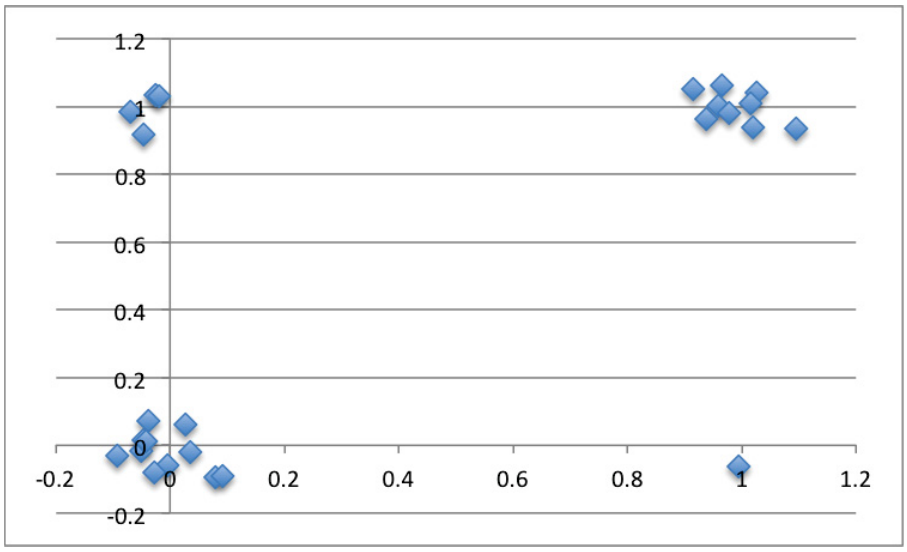

Figure 8.4 RBF (Radial Basis Function) user rating prediction results - 25 instances

In addition, Table 8.13 presents the prediction results on the testing dataset.

For instance, model gives the probability of 0.828 that instance number 1 belongs to the FALSE class. Higher probabilities of FALSE and TRUE for each instance are marked with * in the table.

Table 8.13 RBF (Radial Basis Function) predictions on test dataset

No	Actual	Predicted	Error	Probability Distribution	
				FALSE	TRUE
1	1:FALSE	1:FALSE		*0.828	0.172
2	1:FALSE	1:FALSE		*0.875	0.125
3	1:FALSE	1:FALSE		*0.828	0.172
4	2:TRUE	2:TRUE		0.147	*0.853
5	1:FALSE	1:FALSE		*0.875	0.125
6	2:TRUE	2:TRUE		0.147	*0.853
7	1:FALSE	1:FALSE		*0.828	0.172
8	1:FALSE	1:FALSE		*0.842	0.158
9	2:TRUE	1:FALSE	+	*0.866	0.134
10	2:TRUE	1:FALSE	+	*0.828	0.172
11	1:FALSE	1:FALSE		*0.733	0.267
12	2:TRUE	2:TRUE		0.147	*0.853
13	1:FALSE	1:FALSE		*0.828	0.172
14	2:TRUE	2:TRUE		0.147	*0.853
15	1:FALSE	2:TRUE	+	0.172	*0.828
16	2:TRUE	1:FALSE	+	*0.813	0.187
17	1:FALSE	1:FALSE		*0.875	0.125
18	1:FALSE	1:FALSE		*0.828	0.172
19	2:TRUE	2:TRUE		0.147	*0.853
20	2:TRUE	2:TRUE		0.147	*0.853
21	2:TRUE	2:TRUE		0.147	*0.853
22	2:TRUE	2:TRUE		0.203	*0.797
23	2:TRUE	1:FALSE	+	*0.828	0.172
24	1:FALSE	1:FALSE		*0.828	0.172
25	2:TRUE	2:TRUE		0.149	*0.851

8.1.4.2 Evaluation on test dataset

The RBF classification model correctly classifies the user rating of 80% of 25 applications in the testing dataset. Five applications are misclassified.

Kappa statistic value for this model is 0.6032 which means that there is a good agreement between the model and the actual class.

Table 8.14 presents the RBF evaluation on the testing dataset.

Table 8.14 RBF (Radial Basis Function) evaluation on the testing dataset

Correctly Classified Instances	20	80%
Incorrectly Classified Instances	5	20%
Kappa statistic	0.6032	
Total Number of Instances	25	

8.1.4.3 Detailed accuracy by class

The TP for the FALSE class is 0.917 and is 0.692 for the TRUE class (The greater is, the better). The weighted average TP for the FALSE and TRUE classes is 0.8. The FP for the FALSE class is 0.308 and 0.083 for the TRUE class (The less is, the better). The weighted average FP for the FALSE and TRUE classes is 0.191.

Precision of the SVM classification model for the TRUE is 0.9 and 0.733 for the FALSE. This means that the SVM model predicts the 90% of the TRUE cases exactly and 73.3% of the FALSE cases exactly.

Recall of the model for the TRUE is 0.692 and 0.917 for the FALSE. This means that 91.7% of the FALSE instances are predicted correctly, and 30.8% of the TRUE instances are predicted as FALSE.

Table 8.15 presents the RBF detailed accuracy by class.

Table 8.15 RBF (Radial Basis Function) detailed accuracy by class

	TP	**FP**	**Precision**	**Recall**	**F-Measure**	**ROC Area**	**Class**
	0.917	0.308	0.733	0.917	0.815	0.865	FALSE
	0.692	0.083	0.9	0.692	0.783	0.865	TRUE
WAVG.	0.8	0.191	0.82	0.8	0.798	0.865	

8.1.4.4 Confusion matrix

Table 8.16 presents the RBF confusion matrix. According to this matrix, one of the FALSE cases is predicted as TRUE and four of the TRUE cases are predicted as FALSE.

Table 8.16 RBF (Radial Basis Function) confusion matrix

a	**b**		**<– Classified**
11	1	|	a = FALSE
4	9	|	b = TRUE

8.1.5 SVM (Support Vector Machines)

The dataset is split to 75% for training and 25% for testing. Therefore, there are 74 applications for the training and 25 applications for the testing. The SVM is used to predict the Apple App Store user rating, and results are given in the following sub-sections.

8.1.5.1 Predictions on test dataset

Figure 8.5 presents the Actual against Prediction. Four instances that are misclassified have the X (Predicted class) of zero and Y (Actual class) of one. Remaining instances are placed correctly in the intersection of X and Y axes for zeros and ones. In addition, Table 8.17 presents the SVM predictions on the testing dataset.

For instance, model gives the probability of one that instance number 1 belongs to the FALSE class. Higher probabilities of FALSE and TRUE for each instance are marked with * in the table.

8.1.5.2 Evaluation on test dataset

This SVM model correctly classifies the user rating of 84% of 25 applications in the testing dataset. Four applications are misclassified.

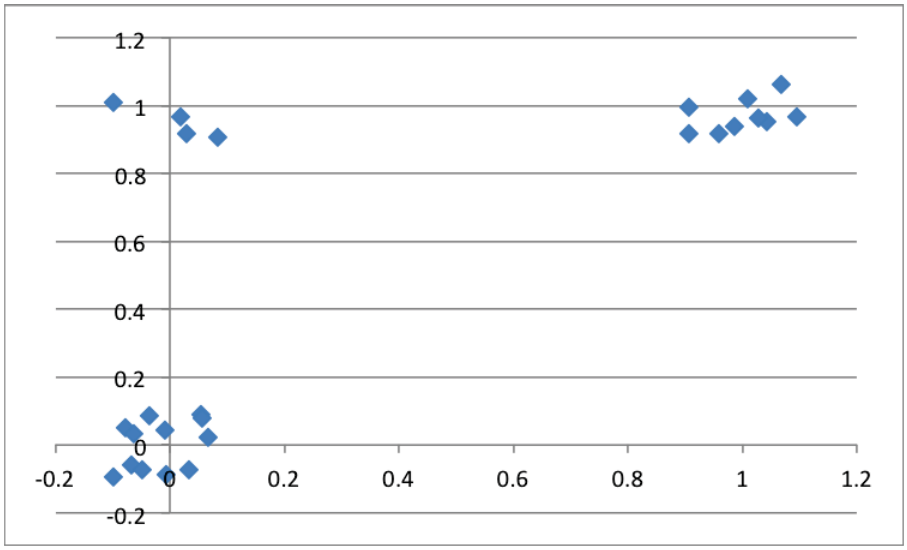

Figure 8.5 SVM (Support Vector Machines) user rating prediction results - 25 instances

Kappa statistic value for this model is 0.6835 which means that there is a good agreement between the model and the actual class.

Table 8.18 presents the SVM evaluation on the testing dataset.

8.1.5.3 Detailed accuracy by class

The TP for the FALSE class is 1.00 and is 0.692 for the TRUE class (The greater is, the better). The weighted average TP for the FALSE and TRUE classes is 0.84. The FP for the FALSE class is 0.308 and 0 for the TRUE class (The less is, the better). The weighted average FP for FALSE and TRUE classes is 0.148.

Precision of the model for the TRUE is 1.00 and 0.75 for the FALSE. This means that the model predicts all of the TRUE cases exactly and 75% of FALSE cases exactly.

Recall of the model for the TRUE is 0.692 and 1.00 for the FALSE. This means that 100% of FALSE instances are predicted correctly, and 30.8% of TRUE instances are predicted as FALSE.

Table 8.19 presents the SVM detailed accuracy by class.

Table 8.17 SVM (Support Vector Machines) predictions on test dataset

No	Actual	Predicted	Error	Probability Distribution	
				FALSE	TRUE
1	1:FALSE	1:FALSE		*1	0
2	1:FALSE	1:FALSE		*1	0
3	1:FALSE	1:FALSE		*1	0
4	2:TRUE	2:TRUE		0	*1
5	1:FALSE	1:FALSE		*1	0
6	2:TRUE	2:TRUE		0	*1
7	1:FALSE	1:FALSE		*1	0
8	1:FALSE	1:FALSE		*1	0
9	2:TRUE	2:TRUE		0	*1
10	2:TRUE	1:FALSE	+	*1	0
11	1:FALSE	1:FALSE		*1	0
12	2:TRUE	2:TRUE		0	*1
13	1:FALSE	1:FALSE		*1	0
14	2:TRUE	2:TRUE		0	*1
15	1:FALSE	1:FALSE		*1	0
16	2:TRUE	1:FALSE	+	*1	0
17	1:FALSE	1:FALSE		*1	0
18	1:FALSE	1:FALSE		*1	0
19	2:TRUE	2:TRUE		0	*1
20	2:TRUE	2:TRUE		0	*1
21	2:TRUE	2:TRUE		0	*1
22	2:TRUE	1:FALSE	+	*1	0
23	2:TRUE	1:FALSE	+	*1	0
24	1:FALSE	1:FALSE		*1	0
25	2:TRUE	2:TRUE		0	*1

Table 8.18 SVM (Support Vector Machines) evaluation on test dataset

Correctly Classified Instances	21	84%
Incorrectly Classified Instances	4	16%
Kappa statistic	0.6835	
Total Number of Instances	25	

Table 8.19 SVM (Support Vector Machines) detailed accuracy by class

	TP	FP	Precision	Recall	F-Measure	ROC Area	Class
	1	0.308	0.75	1	0.857	0.846	FALSE
	0.692	0	1	0.692	0.818	0.846	TRUE
WAVG.	0.84	0.148	0.88	0.84	0.837	0.846	

8.1.5.4 Confusion matrix

Table 8.20 presents SVM confusion matrix results. According to the matrix, all FALSE (Negative user rating) instances are predicted correctly, and four of TRUE (Positive user rating) instances are predicted as FALSE.

Table 8.20 SVM (Support Vector Machines) confusion matrix

a	**b**	**<– Classified**
12	0 |	a = FALSE
4	9 |	b = TRUE

8.2 Machine learning prediction model comparison

If individual developers and development companies were able to predict the user rating of their APP in the Apple App Store before submitting it to the Apple App Store, they could be able to improve their application when the prediction result shows that the user rating would be low.

To be able to predict the APP user rating in the Apple App Store, utilizing a highly accurate prediction model that would employ a dataset of previously evaluated APPs and APPs' user rating in the Apple App Store for the training was necessary.

In this research, a dataset for APP user rating prediction was constructed by evaluating 99 applications. The constructed dataset was used and split into training and testing datasets to avoid overfitting problems and ensure the model validity. Five machine learning models were trained on the training dataset and evaluated on the testing dataset using performance indicators such as precision and kappa statistic.

Table 8.21 presents the classification accuracy, precision for the positive user rating and kappa statistic for each of the five machine learning classification models.

Table 8.21 Comparison of machine learning models

Model	Classification Accuracy	Precision (TRUE)	Kappa Statistic
Naïve Bayes	84%	1	0.6835
Bayesian Network	92%	1	0.8408
MLP	80%	0.9	0.6032
RBF	80%	0.9	0.6032
SVM	84%	1	0.6835

The key observations are:

- The artificial neural network models (i.e. MLP and RBF) have achieved the same classification accuracy (80%), TRUE class label precision (90%) and kappa statistic (0.6032).
- SVM and Naïve Bayes models have achieved the same classification accuracy (84%), TRUE class label precision (100%) and kappa statistic (0.6835).
- Bayesian Network model has achieved 92% classification accuracy, 100% of TRUE class label precision and 0.8408 kappa statistic.

To summarize, this research study has tested the hypotheses proposed in the research objectives section of this thesis by providing an expert-based iOS application usability evaluation model, has used this model to evaluate a subset of 99 APPs, and provided five different machine learning classification models to predict the user rating of these applications in the Apple App Store. All five models were trained on the training dataset and achieved over 80% of accuracy in Positive and Negative user rating prediction on the testing dataset.

Therefore, it is possible to predict an iOS application's user given ratings by evaluating its usability and constructing prediction models. Furthermore, there is a relationship between the usability of an application and its user given ratings in the App Store for this specific dataset and prediction models because all five machine learning classification models could achieve over 80% accuracy in user rating prediction.

8.3 Best model selection

This study selected five different machine learning classification models including, Naïve Bayes, Bayesian Network, MLP, RBF, and SVM. These five models were trained on the same training dataset with 74 instances and 79 criteria and tested on the same testing dataset with 25 instances and 79 criteria.

In summary, the Bayesian Network model has outperformed the other models. The Bayesian Network model has predicted 92% of 25 instances correctly. The precision of the Bayesian Network for the positive user rating in the App Store is one, which means the Bayesian Network model never classified a negative user rating as positive user rating. Indeed, the model has misclassified two TRUE (Actual value) instances as FALSE only. This is not harmful because the objective is to improve the application's usability before publishing it to the App Store. It would be harmful if the Bayesian Network model has predicted a FALSE instance as TRUE (Precision indicator not equal to one) because the developers and development companies would publish an APP to the App Store that is not ready to be published.

Furthermore, the kappa statistic value was 0.8408 for the Bayesian Network model: that means there is an excellent agreement between the Bayesian Network model and the actual class. The fact that Bayesian Network model tested on a dataset other than training dataset, classified 92% of test instances correctly with the precision of 1 for the positive user ratings and has the kappa statistic value of 0.8408 makes it a reliable model.

Thus, the individual developers and development companies can evaluate their applications with the same evaluation model proposed and employed in this research. Next step will be to feed the Bayesian Network prediction model with their evaluation results as a single instance. The prediction model will use the instance as the input and will predict the user rating in the Apple App Store for this instance. Developers and development companies will be able to improve their APP's usability if the Bayesian Network model predicts that user rating will be low, in other words, APP will have a Negative user rating.

To conclude, this study selects the Bayesian Network as the best model among five used models for the used dataset.

8.4 Threats to validity

This section presents the threats to internal and external validity. In this study, a set of 99 APPs has been selected from the App Store. A set of criteria has been provided to evaluate the usability of APPs. Apple iOS characteristics, and the embedded application user experience have been considered in developing the criteria. Changes to the Apple iOS in terms of usability and user experience may cause inconsistency in the criteria. Therefore, the set of criteria should be maintained and kept up to date for future studies.

Furthermore, in this study, three researchers have evaluated 99 APPs according to the criteria that they had developed. These three researchers knew how to interpret the criteria and how to evaluate the applications. The lack of this level of knowledge about how to interpret these criteria may cause a threat to the validity of this research in the future. Therefore, it is very important to understand the iOS platform characteristics and evaluation criteria to be able to evaluate new applications and add them to the dataset.

This study calculates the positive user rating (TRUE class label) and negative user rating (FALSE class label) using the user provided ratings. In some cases, user written comments do not match with user given ratings. Therefore, taking user given ratings only into the account may lead to a wrong conclusion about positivity/negativity of the user ratings.

Moreover, in this study a set of 99 APPs have been evaluated from the App Store and a dataset have been constructed with these evaluations. Furthermore, the dataset has been used to construct the prediction models for the Apple App Store user rating. Prediction model needed to be revisited every time new APPs have been evaluated and added to the dataset. The Bayesian Network model outperformed the other models but by changing the dataset it may not be the case.

Finally, this study did not exhaust all possible classification models to construct the prediction model and only has employed five machine learning models such as Naïve Bayes, Bayesian Network, MLP, RBF, and SVM. There may be other machine learning or non-machine learning models that can predict more accurately the user ratings.

CHAPTER 9

KEY CONTRIBUTIONS AND FUTURE WORK

This chapter presents a summary of the research study, the key contributions of this study and proposes future work.

This study has aimed to construct a prediction model for the Apple App Store APPs (iOS applications) user ratings. To achieve this goal, the following steps have been followed.

Identification of required artifacts related to mobile usability evaluation

This study has reviewed different sources related to mobile application usability evaluation and gathered all the necessary information as:

Usability definition: The definition of usability and its characteristics have been identified in the literature.

Usability in ISO standards: The characteristics and sub-characteristics of usability in ISO 9241 and ISO 25010 standards have been identified.

Guidelines to consider during application design: Design and user experience guidelines as well as the usability heuristics are identified to utilize them in the application usability evaluation.

Mobile human interface guidelines: Mobile human interface guidelines and more specifically Apple HIG have been examined to use the guidelines for the application usability evaluation.

Usability evaluation methods: Usability evaluation methods in the literature have been identified.

Common usability study scenarios: Common usability study scenarios have been identified.

Standardized usability questionnaires: Literature was reviewed for user-based standardized usability evaluation questionnaires.

Usability measures defined by the literature and ISO standards: Example ISO and literature defined usability measures have been identified.

All the obtained information is reported in the chapter 5 of this thesis.

iOS application usability evaluation model construction

An iOS application usability evaluation model was constructed that contains the following artifacts.

- Expert-based iOS application usability criteria for the use of usability evaluation experts
- A user-based iOS application usability questionnaire
- Apple App Store user rating evaluation

Chapter 6 presents the evaluation criteria and Apple App Store user rating evaluation.

Experiments on evaluating a subset of 99 APPs using proposed evaluation model

A set of 99 applications was selected from the Apple App Store. The selected applications have been evaluated against the expert-based iOS application usability criteria by three researchers. The evaluation results were combined with Apple App Store user ratings to construct the dataset.

Chapter 7 presents the experiments for the expert-based iOS application usability evaluation.

Machine learning classification models for prediction of user ratings of Apple App Store iOS applications

Five different machine learning classification models have been used to construct a prediction model with 79 independent variables (i.e. criteria) 1 dependent variable (i.e. user rating class) for the user rating of Apple App Store iOS applications.

Chapter 4 presents the design of the machine learning classification model for the prediction of user rating of Apple App Store iOS applications.

Experiments on iOS application user rating prediction

Five different machine learning models have been utilized to predict the iOS application user rating. The performances of these models have been compared using different performance indicators, and one of the models has been selected as the best model for the dataset.

Chapter 8 presents the experiments on iOS application user rating prediction.

9.1 Key contributions

The key contributions are the followings:

- Proposing an iOS application usability evaluation model and evaluation criteria:
  Systematic literature review has revealed that there was not any publication in the literature specifically targeting the iOS application characteristics. This study has fulfilled the gap by proposing an iOS application usability evaluation model.

- Constructing an Apple App Store user prediction dataset for 99 APPs:
  Systematic literature review has revealed that there was not any dataset in the literature regarding iOS application usability to be used for iOS application user rating prediction. In the course of this study, researchers have done experiments on 99 applications and have constructed the dataset.

- Constructing five different machine learning based user rating prediction models:
  Systematic literature review has revealed that researchers did not consider the Apple App Store user ratings in their studies therefore did not construct a model to predict the user ratings of iOS applications in the App Store. This study has provided a prediction model for the user rating of iOS applications in the Apple App Store.

- Comparing the model performances on the dataset using specific indicators and selecting the best model for the dataset:

This study has compared five machine learning classification model performances with different indicators such as classification accuracy, TRUE class precision and kappa statistic and has selected the best model among the five models for the used dataset.

- Testing if there is a relationship between Apple App Store user ratings and the application usability:

  There are different heuristics and guidelines in the literature that are recommended to be considered during the application design. It was not possible to explore the relationship between these guidelines and Apple App Store user ratings. This study has used a subset of these guidelines to construct a usability evaluation model and its criteria. Furthermore, in the course of this study, researchers have evaluated a set of 99 applications using these criteria and have used the resulting dataset to predict the Apple App Store user ratings. This study presents the results for the user rating prediction and if that there is a relationship between these guidelines and Apple App Store user ratings.

The initial findings of the phase 1 and 2 of this research have been published in the following articles:

- Nayebi, F., J.-M. Desharnais and A. Abran. April 29-May 2, 2012 "The State of the Art of Mobile Application Usability Evaluation", *25th IEEE Canadian Conference on Electrical and Computer Engineering (IEEE CD: 978-1-4673-6/12) Montreal*.

  This paper has been cited by 31 papers according to the Google (Scholar, 2015b) - see Appendix X.

- Nayebi, F.; Desharnais, J.-M.; Abran, A., "An Expert-Based Framework for Evaluating iOS Application Usability, *Joint Conference of the 23rd International Workshop on Software Measurement and the Eighth International Conference on Software Process and Product Measurement (IWSM-MENSURA)*, pp.147,155, 23-26 Oct. 2013 Ankara doi: 10.1109/IWSM-Mensura.2013.30

This paper has been cited by three papers according to the Google (Scholar, 2015a) - see Appendix X.

9.2 Future work

This research study has investigated the user rating problem as a classification problem, in other words as positive user rating (TRUE class label) and negative user rating (FALSE class label). Classes are TRUE if the ratio of fours and fives over ones and twos is over 50%. Future studies can formulate the prediction as a regression problem and try to predict the ratio of four and fives over ones and twos.

This research study did not explore all possible classification models and only presents the results of five classification models such as Naïve Bayes, Bayesian Network, MLP (Multi-Layer Perceptron), RBF (Radial Basis Function) and SVM (Support Vector Machines). Future studies can explore other classification algorithms and compare the results to find the model with better performance.

This research study has proposed an expert-based set of evaluation criteria and a user-based evaluation questionnaire but has only applied the expert-based evaluation. Future studies can evaluate the application with the user-based evaluation criteria and combine them into the dataset.

This research study has proposed the application usability evaluation model for iOS applications and has proposed the user rating prediction for Apple App Store. Future studies can propose these models for other mobile operating systems such as Google Android.

This study calculates the Positive user rating (TRUE class label) and Negative user rating (FALSE class label) using the user given stars only. A sentiment analysis model with text mining techniques (Liu, 2010) could be beneficial to analyze the real sentiment of the users in their comments to come up with the real impression of the users instead of taking only the user rating stars.

Moreover, this study has done the evaluation of 99 applications. Future studies can continue to evaluate more applications and add new instances to the dataset.

CONCLUSION

Smartphones and mobile applications are very popular nowadays because smartphones are portable and provide characteristics that other platforms such as desktop and web cannot provide. Smartphones have integrated cameras for taking pictures, videos, and selfies. They have the GPS functionality that makes them location-aware. They have a built-in microphone and intelligent assistants such as Siri in iOS. They have the integrated calendar, contacts, and storage. Smartphone applications can use any of these characteristics to provide a better user experience and usability. For instance, iOS applications can use the GPS and Map functionality of iPhones to enrich the user experience.

Despite these advantages, smartphones have some characteristics that challenge the usability of their applications. For instance, small screen sizes and different resolutions are required to be considered during application design and development. The information architecture and navigation behavior should be adapted to these small screens.

In addition to the above characteristics, advantages and disadvantages each mobile operating system has its built-in applications and standard user experience. For instance, Apple iOS devices such as iPhones are delivered to the users with a set of built-in applications. All built-in iOS applications follow a set of human interface guidelines. Similarly, Google has a different set of guidelines and user experience for their Android platform.

Hence, different guidelines needed to be considered during the application design and usability evaluation of iOS applications. This study has reviewed the literature on the above characteristics and has revealed that none of the literature studies explicitly considers iOS human interface guidelines and characteristics in usability evaluation.

Because of the popularity of the iOS platform and large number of applications in the Apple App Store, users tend to check the other users given ratings of the applications to examine the appropriate applications. Individual application developers and companies aim to achieve high

user ratings by developing high quality and usability to be successful in the mobile application markets such as Apple App Store.

Predicting the user rating of applications in the Apple App Store before publishing them to the store could be helpful for developers and development companies. This study has reviewed the literature to find a mean to predict the user rating of applications in the App Store. Unfortunately, none of the studies has proposed a model to predict the Apple App Store user rating of applications.

Therefore, this study has aimed to explore and test the following hypotheses:

H1a - Usability of an iOS application and user given ratings in the Apple App Store are related.

All of five proposed Apple App Store user rating prediction models have achieved over 80% of accuracy. Therefore, these models provide strong evidence that there is a relationship between the user rating and the usability of the application in the used dataset.

H0a - There is not any relationship between the usability of an application and its user given ratings in the Apple App Store.

This null hypothesis claims that there is not any relationship between the usability of an iOS application and its user given ratings in the Apple App Store, but in this study the results of the tested hypothesis H1a nullifies this hypothesis.

H1b - It is possible to predict an iOS application's user given ratings by evaluating its usability and constructing a prediction model.

In course of this study five user rating prediction models for iOS applications in the App Store have been developed using usability evaluation and machine learning classification models. The best user rating prediction model has achieved 92% accuracy on the testing dataset.

H0b - It is not possible to predict an iOS application's user given ratings by evaluating its usability and constructing a prediction model.

This null hypothesis claims that it is not possible to predict an iOS APP's Apple App Store user rating, but the testing results of H1b in this study nullifies this hypothesis.

In addition to hypotheses testing and exploration, this thesis has produced the following artifacts:

- This study has reviewed the literature for current mobile usability evaluation methods and has identified the required artifacts for the mobile application usability evaluation.
- ISO and literature defined usability evaluation measures, and standardized questionnaires in the literature have been identified and are presented in this thesis.
- This study has proposed an iOS APP usability evaluation model including expert-based criteria and a user-based usability evaluation questionnaire.
- This study has used 79 expert-based evaluation criteria to evaluate the usability of 99 applications and construct a dataset with the evaluation results. Moreover, the study has provided the statistics and data analysis results regarding this dataset.
- Despite the skewness of the dataset to the left and majority of fours and fives in the dataset, this study has constructed a Bayesian Network machine learning model that could predict the Apple App Store user rating of iOS application with 92% of accuracy.
- This study has employed different machine learning models and indicators to compare them and has selected the best model among them.

Moreover, the following steps have been followed to achieve the objectives of this study.

In phase 1 of this research study, ISO standards, literature, books and standardized usability questionnaires have been identified to have a background knowledge about the domain.

In phase 2 of this research study, an expert-based application usability evaluation model, a user-based iOS application usability evaluation model, and Apple App Store user rating evalu-

ation have been proposed. The meanings of each criterion and the way to interpret them were explained in this phase.

In phase 3 of this research study, an experiment has been conducted to evaluate the usability of 99 applications. This phase has provided the statistics and data analysis results regarding this dataset as well.

In phase 4 of this research study, five machine learning classification models for Apple App Store user rating prediction of iOS applications have been designed.

Phase 5 of this research study has used the dataset built in phase 3 as an input for building five different machine learning based user rating prediction models. Also, the performance of these different models have been compared and the Bayesian Network model that achieves 92% accuracy in prediction has been selected as the best model.

To conclude, this study has proposed a user rating prediction model for iOS applications in the App Store that can be utilized by individual application developers and development companies to predict the application's user rating in the App Store before submitting it to the Apple App Store.

APPENDIX I

THE STATE OF THE ART OF MOBILE APPLICATION USABILITY EVALUATION

Fatih Nayebi[1], Jean-Marc Desharnais[1], Alain Abran [1]

[1] Software Engineering and Information Technologies Depart, École de Technologie Supérieure,

1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

25th IEEE Canadian Conference on Electrical and Computer Engineering (IEEE CD: 978-1-4673-6/12) Montreal, April 29-May 2, 2012

THE STATE OF THE ART OF MOBILE APPLICATION USABILITY EVALUATION

Fatih Nayebi, Jean-Marc Desharnais, Alain Abran

École de Technologie Supérieure – Université du Québec

ABSTRACT

Mobile devices and applications provide significant advantages to their users, in terms of portability, location awareness, and accessibility. A number of studies have examined usability challenges in the mobile context, and proposed definitions of mobile application usability and methods to evaluate it. This paper presents the state of the art of the evaluation and measurement of mobile application usability.

Index Terms—Usability, Quality, Software Measurement, Mobile, Human Computer Interaction.

1. INTRODUCTION

Complex computer systems are finding their way into everyday life, and with a much broader customer base. This has made usability more critical. As a result, companies are seeing the benefits of designing and developing their products with user oriented methods instead of technology oriented methods, and are endeavoring to understand both user and product, by investigating the interactions between them.

Mobile devices and their applications provide significant advantages to their users, in terms of portability, location awareness, and accessibility. Lower price points and improvements in the hardware and software capabilities of smartphones in particular, the so-called "handhelds," have led to tremendous expansion of the mobile and related markets. This has led to huge numbers of mobile applications ("apps") being developed over the past few years.

This vast and increasing number of mobile apps in the marketplace has challenged developers to develop apps of superior quality in order to compete [1]. There are many aspects to the quality of mobile apps, an important one being usability. Furthermore, the architecture of these applications must take into account a number of design constraints, such as limited resources, connectivity issues, data entry models, and the varying display resolutions of mobile devices.

The usability of mobile devices and their apps differs from other computer systems, because their characteristics are different. The software needs of handhelds, such as PDAs and mobile phones, affect the development process of mobile apps, as these are embedded in the phones during manufacturing or installed by customers from various mobile software distribution platforms, such as Apple's App Store and Google's Android Market. Users tend to choose mobile apps that are easy to learn, take less time to complete a particular task, and appear to be more user-friendly because they are less computer-oriented.

In the past, the usability of software systems was evaluated subjectively and the process was not well defined. Researchers would select the aspects of usability to evaluate and measure what they considered important. At the same time, usability measurement and analysis methods and methodologies were being developed. Lab experiments, field studies, and hands-on measurement are some of methodologies most often applied by researchers [2][3][4][5].

Every usability evaluation method has its advantages and disadvantages. Some are difficult to apply, and others are dependent on the measurers' opinions or instruments. In addition to these challenges, mobile devices and applications change very quickly, and updated methods of usability evaluation and measurement are required on an ongoing basis.

This paper presents an analysis of previous studies by considering: 1) up-to-date mobile technologies; 2) the challenges to defining mobile usability questionnaires; and 3) an inventory of measures for mobile usability measurement.

This paper is structured as follows. Section 2 presents some of the generic definitions of usability, as well as specific definitions of mobile usability. Section 3 presents the methodologies for evaluating mobile usability. Section 4 presents observations on the evaluation studies surveyed. Finally, section 5 presents a summary and a discussion on further work.

2. USABILITY DEFINITION

2.1 General definition of usability

It is important to consider the following three aspects of usability for all types of software:

- More efficient to use: takes less time to complete a particular task.
- Easier to learn: operations can be learned by observing the object.
- More user satisfaction: meets user expectations.

ISO 9241 defines usability as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use" [6]. Abran *et al.* [7] consolidate the ISO 9241 [6], IOS 9126 [8], ISO 13407 [9], Dix *et al.* [10], and Nielsen *et al.* [11] usability models, and propose an enhanced one, referred to as the Consolidated Usability Model. This model defines usability as a combination of

effectiveness, efficiency, satisfaction, learnability, and security, along with a recommended set of related measures.

More recently, ISO 25010 [12] breaks down the notion of quality-in-use into usability-in-use, flexibility-in-use, and safety-in-use. In addition, ISO 25010 defines satisfaction-in-use as:

- Likeability: satisfaction of pragmatic goals
- Pleasure: satisfaction of hedonic goals
- Comfort: physical satisfaction
- Trust: satisfaction with security

Also, it defines flexibility-in-use as context conformity-in-use, context extendibility-in-use, and accessibility-in-use.

2.2. Mobile usability definition

With the emergence and rapid deployment of mobile technologies, a number of additional studies have focused on the usability of mobile devices [2][3]: "Problems caused by physical restrictions of mobile devices and wireless networks imply that while designing and conducting usability studies for mobile applications, these issues must be carefully examined in order to select an appropriate research methodology and minimize the potential effect of contextual factors on perceived usability when they are not the focus of studies" [13]. For Zhang *et al.* [13] mobile usability includes some of the new mobility-related challenges, such as: Mobile Context, Connectivity, Small Screen Size, Different Display Resolutions, Limited Processing Capability and Power, and Data Entry Methods.

At the same time, mobile device manufacturers have been enforcing their own usability constraints. For example, the Apple iOS Human Interface Guidelines [14] states the iOS platform characteristics that should be considered during the application development process, such as: Interaction with Multi-Touch screen, Displays of different resolutions and dimensions, Device orientation changes and Gestures such as tap, flick, and pinch. In addition, Apple reviews applications submitted for the App Store based on these characteristics.

Concurrently, Google has developed Android user interface guidelines [15], which guide developers to take into account the following characteristics: Touch gestures, size and location of Icons and Buttons, Contextual Menus and their responsiveness, simplicity, size, and format of Text, and certain aspects of Messages. These guidelines also explain how these characteristics should be considered during the development and testing of Android applications.

In the next section, the methodologies presented in the literature that evaluate the usability of mobile apps are identified.

3. METHODOLOGIES FOR EVALUATING MOBILE USABILITY

Three types of evaluation methodologies are currently used in mobile usability studies:

- Laboratory experiments: human participants are required to perform specific tasks using a mobile app in a controlled laboratory setting.
- Field studies: users are provided with mobile apps and asked about their experience.
- Hands-on measurements: defined aspects of mobile apps are measured directly to evaluate usability.

3.1. Laboratory experiments

Usability laboratory experiments take place in a usability lab where the testing takes place. It is an environment in which users are studied while they interact with a mobile app to evaluate its usability.

By controlling the environment and giving predefined tasks to the users in a usability experiment, it is possible to ensure that they test all aspects of usability. The environment can also be controlled in such a way to isolate users from conditions that are not relevant to the experiment. These advantages make lab experiments useful for comparing different mobile designs and interfaces. In addition, it is possible to record users' activity while they use the mobile apps and analyze the data later.

However, isolating users from environmental factors that can affect usability may cause differences in the user experience, and the effect of environmental factors prevalent in the real world may not be felt. It is also reported that organizing lab experiments is more costly than other methodologies, because of the equipment required [13].

3.2. Field studies

A field study is a general method for collecting data about users, user needs, and product requirements that involves observation and interviews. Data are collected about task flows, inefficiencies, and the organizational and physical environments of users.

Investigators in field studies observe users while they are involved in an activity, taking notes and asking questions. The method is useful early in product development to gather user requirements. It is also useful for studying currently executed tasks and processes. The usability of a mobile app is measured based on participants performing tasks in a real environment. On the downside, sufficient control over users during a field study is not assured.

Some studies have been carried out to develop a questionnaire tailored to measure the usability of electronic mobile products. For instance, the Mobile Phone Usability Questionnaire [MPUQ] was designed in [2] to help researchers evaluate the usability of mobile phones, in order to compare competing devices in the end-user market, and to make decisions among prototypes during the development process and among evolving versions during the iterative design process. Ryu *et al.* [2] group together factors that affect mobile usability, such as: ease of learning and use, the help and problem solving capability provided, emotional aspects and multimedia properties, commands

and minimal memory load, control and efficiency, and typical mobile phone tasks. Additional questions are defined in [2] and linked to these factors.

3.3. Hands-on measurement

There are hands-on measurement methods designed to quantitatively measure the usability of a mobile application, and these require an approach defined by ISO 15939 [16]. This standard explains the measurement process for obtaining base measures using a measurement method, derived measures using a measurement function, and indicators resulting from the analysis of derived measures.

Gafni [3], for instance, in a study of mobile wireless information systems, has developed mobile device-specific usability measures, such as: display load, clarity of operation possibilities, completeness of operation menu, and display self-adjustment possibilities, their purpose and method of calculation. In addition, Gafni links these measures to three types of wireless mobile-related problems: network, device, and mobility.

Hussain *et al.* [4] define a usability metric framework for mobile phone apps, and use the Goal Question Metric approach to link usability goals, such as simplicity, accuracy, and safety, to questions and related metrics.

4. OBSERVATIONS ON THE STUDIES SURVEYED

In this section, we present our observations on the studies we identified that use either a field study or a hands-on measurement methodology. The evaluation studies in which measures and measurement methods are proposed for mobile usability are presented in *Table 1* for articles [2][3][4].

4.1 Limitations of the lab experiments

What Table 1 does not include is reference to evaluation studies based on laboratory experiments. All methodologies have advantages and disadvantages, and one of the most significant criteria is the cost of experimentation. Laboratory experiments need instruments and are therefore more costly than other methodologies. Duh *et al.* [17] compared laboratory and field test methodologies, and concluded the following: "There were many more types and occurrences of usability problems found in the field than in the laboratory. Those problems discovered tend to be critical issues regarding usability. Some of these usability problems are only related to the device being used in the field, which could not be found using conventional laboratory usability tests. With regards to the users' behaviors, users behave less positively and more negatively in the field than in the laboratory. Some behaviors can only be observed in the field. Users also take longer time to perform certain tasks and also present more negative feelings, such as dissatisfaction and difficult of use, to the use of the device in the field." This is the other reason why this methodology is not considered.

4.2. New mobile user interface needs

We observed in the literature that the questionnaires and hands-on methods developed for mobile usability measurement do not consider the user interface features provided in the newest mobile operating systems that are gaining popularity. The multi-touch gestures (e.g. the tap, flick, and pinch), device orientation changes, and location awareness are not examined.

4.3. Usability measures

The measures [16] discussed in the identified studies are often not properly defined. For example, it is not clear on what basis users are answering questions posed in questionnaires [2], such as, "Does the product have all the functions and capabilities you expect it to have?" This question is imprecise, and so the answer will be subjective and highly dependent on the user's judgment.

4.4. Measurement methods

Sometimes the measurement methods [16] are not properly explained for some of the hands-on data collected. For instance, it is not clear how what measurement methods were used to collect data such as: "Number of system resources displayed," "Number of voice assistance in a task" [4].

4.5. Single evaluation methodology

Published studies typically rely on a single methodology. However, a field study, along with hands-on measurement, could be more informative in mobile usability measurement studies. In this case, the field study is based on the experience of users, while hands-on measurement is performed by a measurement professional. Combining the two methodologies should provide more significant information for evaluation.

4.6. Mobile market rates

User given rates in app markets, such as Apple's App Store and Google's Android Market, are valuable resources for usability studies; however, the studies we examined did not consider mobile marketplace ratings.

Table 1: Summary of evaluation studies

Reference number	2	3	4
4.2. New mobile user interface needs	N	N	N
4.3. Usability measures	N	A	A
4.4. Measurement methods	N	A	P
4.5. Single evaluation methodology	N	N	N
4.6. Mobile market rates	N	N	N

A: available N: not available P: partially available

Furthermore, there is as yet no standardized set of measures and corresponding measurement methods covering popular mobile operating systems, which can be used for benchmarking purposes to compare the usability of mobile apps.

5. SUMMARY AND DISCUSSION

In this study, we looked at the mobile application usability evaluation and testing, and determined that there is no scientific research addressing the requirements of new mobile user interfaces.

We also surveyed the literature on the usability measures and measurement methods used in both field studies and hands-on measurement, and also determined the importance of mobile market rates. A study of the literature also revealed that there are no evaluation studies related to iOS mobile application usability measurement.

Future work is needed to investigate the following issues:

1) iOS Human Interface Guidelines aligned measures: a vast number of applications have been developed for this platform, and it is Apple's objective to review all applications before publishing them in their App Store. However, this is a time-consuming process, and self-review before submitting an application to the store is clearly needed.

2) A field study methodology is needed that takes the form of a questionnaire that can consider new mobile operating system needs, such as interaction with a multi-touch screen, displays of different resolutions and dimensions, device orientation changes, and gestures like tap, flick, and pinch.

3) A hands-on measurement methodology is needed that can be applied to the same application development process as considered in the field study.

Furthermore, it is necessary to define base and derived measures and their measurement methods for hands-on measurement. Defining these elements will give practitioners the ability to compare and more precisely benchmark the usability of the mobile applications.

4) Finally, there is a great deal of data on user satisfaction on the mobile application markets that can be used for analysis. In addition, these markets can help researchers collect user feedback from all over the world. At the same time, by distributing the developed applications in mobile markets, researchers can prototype and evaluate their research methodology.

6. REFERENCES

[1] Global mobile statistics 2011, http://mobithinking.com/mobile-marketing-tools/latest-mobile-stats

[2] Y. S. Ryu, "Development of usability questionnaires for electronic mobile products and decision making methods," Dissertation Submitted to the Faculty of Virginia Polytechnic Institute and State University in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Industrial and Systems Engineering, 2005.

[3] R. Gafni, "Usability issues in mobile-wireless information systems," Issues in Informing Science and Information Technology, vol. 6, 2009, pp. 755-769.

[4] A. Hussain and M. Kutar, "Usability Metric Framework for Mobile Phone Application," The 10th Annual PostGraduate Symposium on The Convergence of Telecommunications, Networking and Broadcasting, June 22-23, 2009.

[5] C. K. Coursaris and D. J. Kim, "A Meta-Analytical Review of Empirical Mobile Usability Studies," Journal of Usability Studies, vol. 6, no. 3, pp. 117-171, 2011.

[6] ISO/IEC 9241 Ergonomics requirements for office with visual display terminals (VDTs), International Organization for Standardization, Geneva, Switzerland.

[7] A. Abran, A. Khelifi, W. Suryn, and A. Seffah, "Consolidating the ISO usability models," Proceedings of 11th International Software Quality Management Conference and the 8th Annual INSPIRE Conference, pp. 23-25, 2003.

[8] ISO/IEC 9126 Software Product Evaluation -- Quality Characteristics and Guidelines for the User, International Organization for Standardization, Geneva, Switzerland.

[9] ISO/IEC 13407 Human-centered design processes for interactive systems, International Organization for Standardization, Geneva, Switzerland.

[10] Dix, A., Finlay, J., Abowd, G., and Beale, R. 1993. Human-Computer Interaction, Prentice-Hall, NJ, USA.

[11] Nielsen, J. 1994. Usability Engineering. Boston: Academic Press.

[12] ISO/IEC 25010:2011 Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models, International Organization for Standardization, Geneva, Switzerland.

[13] D. Zhang and B. Adipat, "Challenges, methodologies, and issues in the usability testing of mobile applications," International Journal of Human-Computer Interaction, vol. 18, no. 3, pp. 293-308, 2005.

[14] Apple iOS Human Interface Guidelines: http://developer.apple.com/library/ios/#documentation/UserExperience/Conceptual/MobileHIG/Introduction/Introduction.html

[15] Google User Interface Guidelines: http://developer.android.com/guide/practices/ui_guidelines/index.html

[16] ISO/IEC 15939:2007 Systems and software engineering – Measurement process, International Organization for Standardization, Geneva, Switzerland.

[17] Been-Lirn Duh, H., Tan. G. C. B. Usability evaluation for mobile device: A comparison of laboratory and field tests. Proc. 8th International Conference on Human Computer Interaction with Mobile Devices and Services, September 12-15, 2006, Espoo, Finland.

APPENDIX II

AN EXPERT-BASED FRAMEWORK FOR EVALUATING iOS APPLICATION USABILITY

Fatih Nayebi[1], Jean-Marc Desharnais[1], Alain Abran [1]

[1] Software Engineering and Information Technologies Depart, École de Technologie Supérieure,

1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

"Joint Conference of the 23rd International Workshop on Software Measurement and the Eighth International Conference on Software Process and Product Measurement (IWSM-MENSURA)", pp.147,155, 23-26 Oct. 2013 Ankara doi: 10.1109/IWSM-Mensura.2013.30

An Expert-based Framework for Evaluating iOS Application Usability

Fatih Nayebi, Jean-Marc Desharnais and Alain Abran
Software Engineering and Information Technologies Department
École de Technologie Supérieure, University of Quebec, Montreal, QC, Canada
fatih.nayebi.1@etsmtl.net, jean-marc.desharnais@etsmtl.net and alain.abran@etsmtl.ca

***Abstract*—Mobile applications are gaining in popularity because of the significant advantages of mobile devices, such as portability, location awareness, electronic identity, and an integrated camera. However, these devices have a number of disadvantages in terms of usability, like limited resources and small screen size. Evaluating the usability of applications developed for mobile operating systems is a very important step in addressing these disadvantages and achieving success in mobile application markets, such as Apple's App Store. Usability evaluation must be tailored to all the various mobile operating systems in use, as they each have their own particular characteristics. This paper presents a mobile application usability evaluation framework for one of the most popular mobile operating systems, iOS. A set of critera is defined and applied to evaluate the usability of eleven applications available at the App Store.**

Keywords*—*Mobile Human Computer Interaction, Application Usability Evaluation, iOS App Design and Development

I. Introduction

There are two principal methods of usability evaluation, depending on who participates in the evaluation: the user in a user-based evaluation, and the expert in an expert-based usability evaluation.

In user-based evaluations, users with different profiles and from different backgrounds try to complete predefined tasks. These evaluations are divided into two major types: laboratory experiments, and field studies.

In laboratory experiments, an application (or app) is used in a prepared environment. The session is recorded by cameras, and the recordings are examined by usability experts [1]. In field studies, usability experts provide an app to users, along with a related questionnaire to capture their perceptions of their experience with the app.

User-based evaluations are important because they measure the perceptions of an app by the end-user, whose opinion matters a great deal. However, they do not measure the app based on the principles, guidelines, and heuristics that should be considered during the app design process. For example, the principles that underlie apps are detailed, and evaluating them requires expert knowledge.

This study focuses on the expert-based usability evaluation of iOS apps. We examine various well-known heuristics, principles, and guidelines from the literature in the section 2. The term guidelines in this study refers to the concepts to be considered in expert-based evaluation, and they are defined here in the form of criteria designed to quantify usability.

Mobile devices and operating systems (iOS in this study) have their own characteristics that should be considered during app design, and later for usability evaluation.Apple provides iOS Human Interface Guidelines (HIG) [2] specifically for iOS apps, and reviews apps submitted to the App Store based on these guidelines.

The iOS HIG cover platform characteristics, human interface principles, and user experience guidelines, and experts are needed to examine all these aspects of app design in a usability evaluation process before an app is submitted to the App Store. We look at the HIG and define questions associated with each of them to enable usability quantification.

This paper is structured as follows. Section 2 explains the heuristics to be considered during app design, and the related questions for usability evaluation. Section 3 presents the iOS HIG themselves, and the questions formulated for the evaluation. In section 4, an evaluation example is presented. Finally, section 5 concludes the study and suggests future work.

II. Guidelines to consider during app design

App designers consider specific guidelines to make apps as usable as possible, and usability experts examine apps based on these guidelines to determine the extent to which the design complies with them. Of course, guidelines are sometimes subjective, and the solutions they provide are dependent on the examiner's judgment. For this reason, a standardized process and measures for expert-based usability evaluation based on these heuristics are important, both for benchmarking and for comparing different apps.

This section presents user interface (UI) design guidelines to consider during the app design process. The aim of this study is to propose a framework of guidelines for evaluating the usability of an iOS app, along with the criteria to interpret them. We examine these guidelines (which are listed in Figure 1), and define questions for evaluating the extent of compliance of apps with them. In later subsections, we explain the guidelines and define questions to quantify them.

A. Usability heuristics

Usability heuristics are rules of thumb which should be followed by all UI designers. They can also be considered by usability examiners during usability evaluation. Sets of usability heuristics have been developed by Nielsen [3], Gerhardt-Powals [4], Shneiderman [5], and Weinschenk and Barker

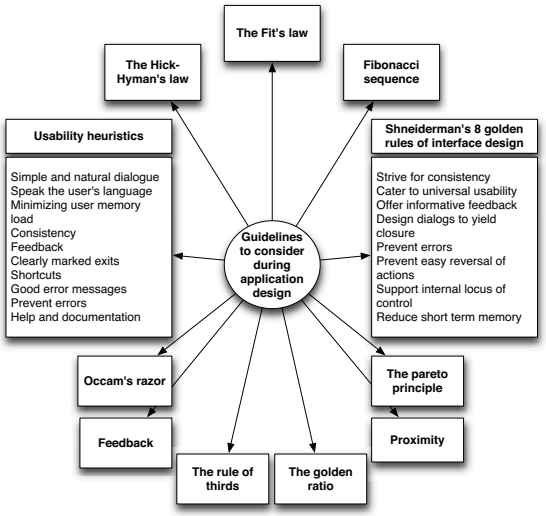

Fig. 1. Guidelines to consider during application design

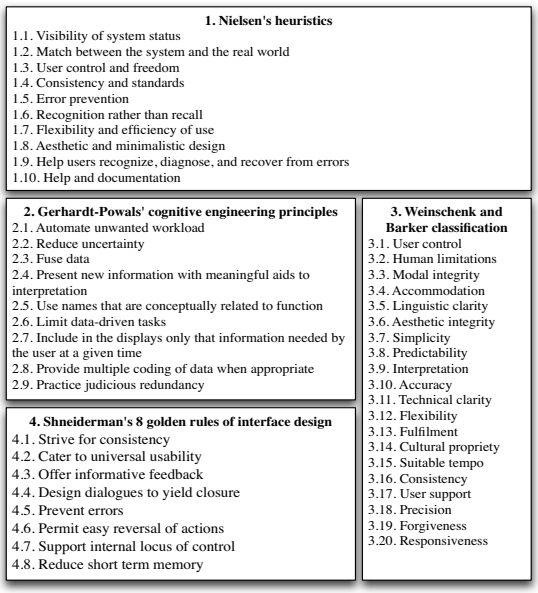

Fig. 2. Usability heuristics

[6] – see Figure 2. In this study, we examine these sets of usability heuristics, and select a subset of them for the usability evaluation of iOS apps, and define questions to evaluate an app's level of compliance with the heuristics we have chosen. The answers to the questions are multiple choice, and quantified on an ordinal scale (from 1 to 5 in this case).

The subset of usability heuristics that will be used in this study includes the following:

1) User Control and freedom: The UI provides control and freedom for the user of the app.

(a) The user can leave an unwanted state via clearly marked cancel/exit points.
(b) The app supports undo and redo.
(c) The user can initiate and control the pace and sequencing of the interaction.
(d) The user knows where he is in the app, how he got there, and where he can go via a navigation controller stack and accurate view names.
(e) The user is kept informed of his current position and of the number of steps he must go through to reach his goal.

2) Error correction: Error correction messages should express the problem precisely, in plain language, and suggest a solution.

(a) When an error occurs, the app tells the user what happened, why, and how to fix it.
(b) Required fields are made clear to the user via visual indicators.
(c) A back button or gesture returns the app to a previous view without loss of data.

3) Human Limitations: The design takes into account human limitations, both cognitive and sensory, and helps the user via memory support and appropriate instructions.

4) Accommodation: The design meets the needs and behaviors of all targeted user groups, and is user-friendly.

(a) The language of the app embodies words, phrases, and concepts which are familiar to the user.
(b) Relevant metaphors representing real-life objects are used when needed to help the user understand, and learn, the task.
(c) The UI is appropriate for the user's task and skill level, enabling him to focus on the task itself, rather than on the technology chosen to perform the task.

5) Linguistic Clarity: The language used in the app is clear, effective, and appropriate for the audience.

(a) The app contains no spelling or grammatical mistakes, which could damage the user's trust.
(b) Abbreviations and acronyms are not used in an app, unless they are straightforward and easily understood.

6) Aesthetic integrity: The design is visually appealing and well integrated into the functionality of the app, providing similarity, continuity, completion, proximity, and figure/background differentiation.

(a) Similarity occurs when objects look similar to one another, and can be perceived as part of a group or pattern.
(b) Continuity occurs when the eye is compelled to move from one object another.
(c) Completion occurs when an incomplete object or a space that is not completely filled can be perceived by the user as a whole when he adds the missing information.
(d) Proximity occurs when elements are placed close together, and can be perceived as belonging to a group.
(e) Figures (forms, silhouettes, and shapes) are differentiated from background (the surrounding area). Balancing figures

and background can make the perceived image clearer. Unusual figure/background relationships can add interest and subtlety to an image.

7) Simplicity: Simplicity refers to the tailoring of the design to appeal to the targeted user groups, and to the avoidance of unnecessary complexity, owing to the restricted screen size. The following criteria are aimed at evaluating the simplicity.

(a) Views contain information that is relevant or needed.
(b) Color coding is used for clarity where appropriate.
(c) The size of graphics is optimized for performance and proper resolution for response time impact.
(d) The number of colors is limited to 3-4.
(e) The app's purpose and usage area can be readily understood from the start.

8) Predictability: The user will be able to predict the behavior of the system in response to his actions.

(a) The app makes it clear which dialog the user is in, where he is in the app, and which actions he can take and how to perform them.

9) Flexibility: The design is flexible enough to adapt to the needs and behavior of the user.

Shortcuts may speed up the interaction between the user and the app, and may also increase user control over the app.

(a) The app allows the user to work in a way that suits him.
(b) The user doesn't need to use workarounds or manuals.
(c) The most frequently used parts of the app are listed from top-left to bottom-right, in order of importance.
(d) The UI can be individualized by the user to modify his interaction with the app and have the information presented in a way that suits his capabilities and needs.
(e) Tappable and untappable areas of the app are clearly recognizable.
(f) Shortcuts have been developed for the most frequently used parts of the app.
(g) The navigation points between starting locations and tasks and are clear to the user.

10) Consistency: The styles of the various parts of the system are consistent.

Users shouldn't have to wonder about the meaning of words, situations, or actions in an app. There should also be consistency in the interactions between the user and the app, and in the use of gestures.

(a) The UI conforms to the user's expectations, in that it meets the predictable contextual needs of the user and respects commonly accepted conventions.
(b) All the views are displayed consistently, so that users can apply knowledge gained in one part of the app to the system as a whole.
(c) Labels and titles are consistent throughout the app, and accurately define the tasks to be performed in the app.
(d) The app's interactions and gestures meet the expectations of the user, in that they are standard and predictable.

11) User Support: The design of the app supports learning and provides the required assistance to the user.

It is best if an app can be used without any help or documentation. If it is needed by the user, the app should provide a list of concrete steps to accomplish the tasks.

(a) The app provides easily accessible help to users when needed.
(b) The Help documentation is properly prepared, and is both appropriate and informative.
(c) The user can easily move between Help and the current task.
(d) Help does not interfere with the task flow.
(e) Help is context-based, and addresses all the necessary contexts.

12) Forgiveness: The app 'forgives' the user for committing an error, and enables him to recover successfully by providing clearly marked exit points.

(a) The UI is error-tolerant, with error management strategies in place to deal with errors without the need for action by the user.
Even better than good error management is a careful design that prevents a problem from occurring in the first place.
(b) The app makes it difficult to make mistakes by preventing them with items like a confirm command.
(c) The app validates the information that the user enters into data forms, informing him if it is not in an acceptable format.

13) Responsiveness: The UI is responsive, in that it provides sufficient and timely feedback about the app's status and signals task completion.

(a) The app keeps the user informed about the send/receive status of content via a progress indicator.
(b) The UI supports Undo and Redo.
(c) The app enables the user to leave an unwanted state without having to embark on an extended UI interaction.

B. Hick-Hyman's law

References [7] and [8] state that the time it takes to make a selection increases with every additional choice available. This means that the more options a user has when using a mobile app, the more difficult it will be to use. This underlines the importance of simplicity.

Given n equally probable choices, the average reaction time T required to choose among them is approximately

$$T = b * \log_2(n+1) \tag{1}$$

where b is a constant that can be determined empirically by fitting a line to the collected data. This logarithmic operation expresses depth of the choice tree hierarchy. Basically $log2$ means that a binary search is performed. The law can be generalized to the case of choices with unequal probabilities p_i occurring, with

$$T = bH \tag{2}$$

where H is the information-theoretic entropy of the decision, defined as

$$H = \sum_{i}^{n} p_i \log_2(1/p_i + 1) \tag{3}$$

where p_i refers to the probability of the i^{th} alternative yielding the information-theoretic entropy.

The implication here is that designers should minimize the number of choices a user has. Removing unnecessary pages, links, buttons, or selections will make designs much more effective.

C. Fitt's law

According to Fitt, "the time required to move to a target is a function of the target size and distance to the target." [9]. Designers can apply this to app design by looking at the hit area of the objects used, meaning that the larger the tappable area of the navigational elements, the easier it will be to tap.

Fitts' guideline has been formulated mathematically in several different ways. A common form is the Shannon formulation (proposed by MacKenzie, and named for its resemblance to the Shannon-Hartley theorem) for movement along a single dimension:

$$T = a + b * \log_2(1 + D/W) \tag{4}$$

where:

- T is the average time taken to complete the movement. (Traditionally, researchers have used the symbol MT for this, which stands for movement time.)
- a represents the start/stop time of the device (intercept)
- b stands for the inherent speed of the device (slope). Constants a and b can be determined experimentally by fitting a straight line to the measured data.
- D is the distance from the starting point to the center of the target. (Traditionally, researchers have used the symbol A for this, which stands for the amplitude of the movement.)
- W is the width of the target measured along the axis of motion. It can also be thought of as the error tolerance allowed in the final position, since the final point of the motion must fall within $W/2$ of the target's center.

From the equation, we can see that a speed-accuracy tradeoff is associated with pointing, that is, targets that are smaller and/or further away require more time to acquire.

D. Fibonacci sequence

The Fibonacci sequence is a series of numbers in which each number is the sum of the preceding two [10]. For example, starting with 1, the sequence is the following:

$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, etc...$

The Fibonacci sequence is one of the most important patterns in both mathematics and design, and this has been recognized in many classical works. It is commonly found in nature, and, like the golden ratio (see below), it is often used to create visual patterns, shapes, and organic figures, to build grids, and to dictate sizing and ratios. Moreover, patterns based on this sequence are intrinsically esthetic, which is an added advantage to its use in design composition.

E. Occam's razor

According to Occam's razor, "the simplest solution is almost always the best" [11]. With the flexibility and power of the mobile device and its design tools, it is easy to be carried away in the design process. The result could be a very complicated app or design with a great deal of functionality and information, but difficult to build, use, and maintain. The app may promise more, but actually achieves less.

This is commonly an issue when a company feels the need to put everything they possibly can into a mobile app, in the belief that someone will want the information. What such a company fails to consider is that the overwhelming majority of users will access only about 20% of the content.

Being ruthless about the value that a view or piece of content provides and removing any content that is unnecessary will result in significantly stronger and more effective designs.

Occam's razor also implies that "a design isn't finished when there is nothing more to add, but when there is nothing left to take away." Design simplicity is elegant, sophisticated, and much more effective than the complex decorative style that is so prevalent in mobile apps.

F. Pareto guideline

According to the Pareto guideline, a high percentage of users will perform a low percentage of actions [12]. This means that most users will go to a small number of views. In the case of mobile apps, most users will also perform a small number of tasks.

Based on this guideline, a designer can identify what actions most users will perform (from analytics, research, interviews, etc.), and then puts greater emphasis on those tasks and actions to make the app easier to use. Sometimes this can involve the inclusion of a new navigation feature, or modification of the homepage to make finding and accomplishing the tasks easier.

It can also involve paring down and removing content and features from a mobile app. If users are not accessing or using the information, then the usability of the application can be improved by removing it. This leads us back to Hick's Law and Occam's Razor.

G. Rule of thirds

The rule of thirds is a method of composing elements to make them visually pleasing, as well as to identify ways in which the user's eyes will scan the view [13]. Photographers have been using this heuristic for years to create more visually interesting compositions.

The rule of thirds operates by breaking up a design into thirds, both vertically and horizontally, and building a grid of intersecting lines. According to this rule, a viewer is more

likely to be drawn to the intersections than to the lines or the spaces between them. Good rules of thumb are to place elements along the lines and at the intersections, and to avoid placing anything in the dead center of the composition or having a horizon dividing the composition in half.

Placing elements so that they take up one-third or two-thirds of the space will be more visually pleasing to most viewers.

H. Golden ratio

The golden ratio is often confused with the rule of thirds. In fact, this ratio looks at what proportions are naturally the most visually appealing [14]. It has been used in design, architecture, and engineering for hundreds of years.

The golden ratio is calculated using the elements of a shape, such as height to width, and turns out to be approximately 0.618.

When applied to rectangles, the golden ratio can be used by a designer to break the shape down into smaller renderings, creating a natural spiral pattern. This pattern can be seen in nature by examining sea shells.

I. Proximity

Elements close to one another will appear related [15], a fact that is often overlooked.

What this means is that a designer must be aware of how much space he is placing between the elements of a design. If the elements of a series are too close together, the user will assume that this has been done intentionally, and that the elements are related. This is often an issue with mobile apps, where buttons or controls are grouped together, and yet have unrelated functionality. The result is confusion on the part of the user, who is trying to understand the app.

III. iOS Human Interface Guidelines

Apple's iOS Human Interface Guidelines [2] list the iOS platform characteristics that should be considered during the app development process, such as interaction with a multi-touch screen, different display resolutions and dimensions, device orientation changes, and gestures like tap, flick, and pinch. The company reviews apps submitted to the App Store based on these characteristics.

The Human Interface Guidelines (HIG) [2] are intended to help designers and developers build the highest quality UIs for their iOS apps and offer the best possible user experience [iOS HIG]. Apple claims that working with the platform conventions will better position developers to create more user-friendly iOS apps. Figure 3 presents the characteristics of the platform, as well as the human interface principles and the user experience guidelines from the HIG document. We describe and explain the user experience guidelines that are the most relevant to this study, and propose questions that can be posed for quantifying the level of compliance with the guidelines.

According to the Human Interface Guideline, "The user experience of iOS-based devices revolves around streamlined interaction with content that people care about". The following guidelines apply to apps that run on all iOS-based devices.

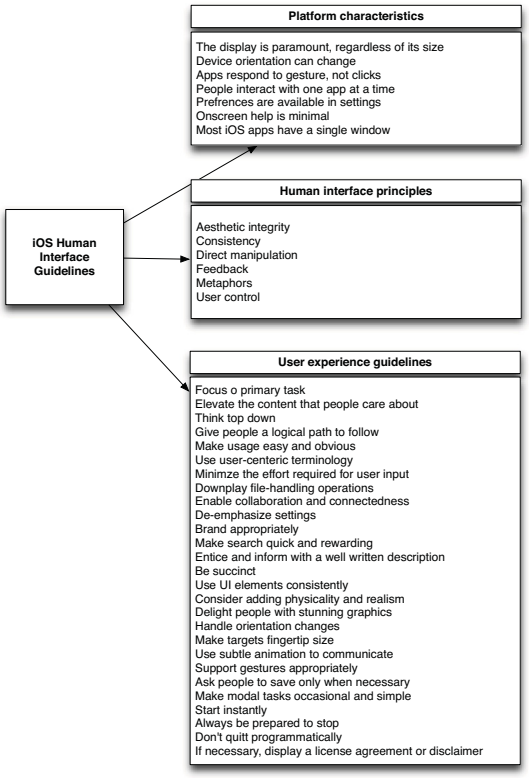

Fig. 3. iOS Human Interface Guidelines

A. Focus on the Primary Task

Focus on the primary task, determining the most important content for each context or screen.

B. Elevate the Content that People Care About

Elevate the content that users care about, by designing the app's UI as a subtle frame for the information they are interested in; for example:

(a) Minimize the number and prominence of controls, in order to decrease their weight in the UI.
(b) Subtly customize the controls, so that they integrate with app's graphical style, and can be discovered and understood without being conspicuous.
(c) Fade controls for a little while after the user has stopped interacting with them, and display them again when the user taps the screen, to help him concentrate more

C. Think Top Down

The top of the screen is the most visible to users and the easiest to reach, because they tend to hold the device in their hands; for example:

(a) Place the most frequently used (usually higher level) information near the top, and in the following order: from general to specific, and from high level to low level.

D. Give People a Logical Path to Follow

Give users a logical path to follow, so that they know where they are in the app and receive confirmation that they are on the right path.

(a) The path should be predictable, and markers, such as the back button, should be provided to inform users where they are and how to retrace their steps.
(b) Provide only one path to a screen in most cases; if a screen needs to be accessible in different circumstances, provide a modal view as well.

E. Make Usage Easy and Obvious

Strive to make the app instantly understandable to users, so that they don't have to spend time figuring out how it works; for example:

(a) Make the main function of the app immediately apparent, by minimizing the number of controls from which the user must choose, and use standard controls and gestures appropriately and consistently, so that they behave the way the user expects them to.
(b) Make the app consistent with the usage paradigms of built-in apps, with the same screen navigation hierarchy, content listing style, and mode switching capability using the tab bar.

F. Use User-Centric Terminology

User-centric terminology should be used in applications. for example:

(a) Use understandable terminology, that is, words and phrases that are appropriate for the targeted user groups, in all text-based communications.
(b) Describe dates accurately in the UI, avoiding informal terms such as "today" or "tomorrow", which may not reflect the user's current location.

G. Minimize the Effort Required for User Input

Minimize the effort required for user input, as it is time-consuming and requires the user's attention, whether it is tapped in or keyed in; for example:

(a) Balance the user input requested with what the app offers the user in return, providing as much information or functionality as possible for each piece of information entered by the user.
(b) Make choices easy for the user, e.g. by providing a table view or a list picker component instead of a text field.
(c) Obtain information from the device when it makes sense to do so, so that users aren't obliged to provide information that is easily accessible by the app.

H. Downplay File-Handling Operations

Downplay file handling operations, so that users are minimally aware of the existence of a file system on an iOS device, unless they are creating or manipulating files themselves.

Specifically, users should not be encouraged to think about file metadata or locations.

(a) Ensure, to the extent possible, that users can manage documents without opening iTunes on their computer, by using iCloud, for example, to help them access their content on all their devices.

I. Enable Collaboration and Connectedness

Enhance the app by enabling users to collaborate and connect with others in accordance with the mission of iOS devices, which, even though they are personal devices, encourage sharing with others; for example:

(a) Ensure that users are able to easily share information that is important to them, like their location, opinions, and high game scores, when it is appropriate.

J. De-emphasize Settings

(a) Include settings about preferred app behaviors and information that users rarely want to change in the app only when it is appropriate to do so.
(b) Allow users to set their preferred behaviors by using the configuration options in the app.
(c) Offer configuration options in the main UI, or on the back of a view in iPhone apps: to decide which location makes sense, and to determine whether or not the options involve a primary task and how often users might want to set them.

K. Brand Appropriately

(a) Present the brand colors or images in a subtle and understated way for greatest effect.
(b) Avoid taking space away from the content that users care about, and using it to display branding assets.

L. Make Search Quick and Rewarding

Make searching quick and rewarding, by making the search function the primary one, and following the guidelines below to ensure that it performs well; for example:

(a) Build app data indices, so that the app is always ready to be searched.
(b) Live-filter local data, so that the app can display results more quickly, narrowing them as the user continues to type.
(c) Filter remote data while the user types when possible, informing him that he can opt out if the response time is likely to delay the results by more than a second or two.
(d) Display the search bar above a list, or the index for a list, which is where users are accustomed to finding it in MS Outlook Contacts and other apps.
(e) Feature the search function as a distinct mode if it is a primary function in the app, and only provide a search tab in special circumstances.
(f) Display placeholder content and partial results as they become available to give users prompt access.
(g) Provide a scope bar if the data sort naturally into different categories, as this allows users to specify locations or rules in a search, or to filter objects by specific criteria.

M. Entice and Inform with a Well-Written Description

An App Store description is a great opportunity to communicate with potential users. In addition to accurately describing the app and highlighting the qualities users might appreciate the most, follow these guidelines:

(a) Ensure that all spelling, grammatical, and punctuation errors are corrected, to avoid creating a negative impression of an app's quality.
(b) Keep all-capital-letter words to a minimum, as they can make text very difficult to read.
(c) Describe specific bug fixes that customers have been waiting for in the description of a new version of an app.

N. Be Succinct

(a) Convey information in a condensed, headline-type style, so that users can absorb it quickly and easily.
(b) Give short labels or well-understood symbols to controls, so that users know what they are doing at a glance.

O. Use UI Elements Consistently

Use UI elements consistently, as users expect standard views and controls to look and behave in the same way across apps; for example:

(a) Follow usage recommendations for standard UI elements, so that users can depend on previous experience to help them learn to use a new app.
(b) Avoid radically changing the appearance of a control that performs a standard action, as users will spend time discovering how to use them and wonder what, if anything, this control does that the standard one does not.
(c) Never use the standard buttons and icons to mean something else.

P. Consider Adding Physicality and Realism

Consider adding a physical, realistic dimension to the app when appropriate; for example:

(a) Enhance scenes or enlarge objects, if appropriate, to communicate with users and to express the essence of the app, as this can convey more meaning than a faithful likeness.
(b) Use appropriate animation to enhance realism in an app, bearing in mind that users will accept artistic license in appearance, but they may feel disoriented when they see movement that appears to defy the laws of physics.

Q. Delight People with Stunning Graphics

Delight users with rich, beautiful, and engaging graphics, as they draw users into the app and make the simplest task rewarding, as well as helping to build the app's brand; for example:

(a) Replicate the look of high-quality or precious materials, taking the time to make sure the material looks realistic and valuable.
(b) Create beautiful, high-resolution artwork and icons from the start, rather than scaling up the quality later.

R. Handle Orientation Changes

Allow orientation changes, as these are often expected by iOS device users:

(a) Maintain focus on the primary content in all orientations, so that users feel they have control over the app and the content they care about.

S. Make Targets Fingertip-Size

Make targets fingertip-size to ensure that users can comfortably use the app, following these guidelines:

(a) Tappable elements in an app have a target area of about 44 x 44 points, as this size is important for ease of use.

T. Use Subtle Animation to Communicate

Use subtle animation to communicate, because it is highly effective, as long as it doesn't get in the way of the users' tasks or slow them down; for example:

(a) Keep animation consistent with that built into iOS apps when appropriate, as users are accustomed to the subtle animation they contain.
(b) Use animation consistently throughout the app, so that users can rely on the experience it gives them.

U. Support Gestures Appropriately

Support gestures appropriately and predictably, as iOS device users use gestures to interact with their iOS devices and associate certain behaviors with specific gestures; for example:

(a) Avoid changing the actions associated with the standard gestures that users know.
(b) Assign complex gestures, or less common ones like swipe or pinch open, as shortcuts to expedite a task, not as the only way to perform a task.

V. Ask People to Save Only When Necessary

Ask users to save only when necessary, as the app, not the user, should save the data, and do so automatically.

W. Make Modal Tasks Occasional and Simple

Make modal tasks simple and infrequent, minimizing, when possible, how often the user must be in a modal environment to perform a task or provide a response.

(a) Always provide an obvious and safe way to exit a modal task, to reassure users that their work is safe when they dismiss a modal view.

X. Start Instantly

Present useful content immediately, as it is often said that users spend no more than a minute or two evaluating a new app.

(a) Display a launch image which closely resembles the first screen of the app, to decrease the app's perceived launch time.

(b) Avoid displaying an About window or a splash screen, to ensure that users are not prevented from using the app immediately.
(c) Delay the login requirement for as long as possible, to enable users to navigate through much of the app and access some of its functionality without logging in.
(d) Ensure that, when an app restarts, its state is restored, so that users don't have to remember how they had reached it in the first place.

Y. Do not quit programmatically

Try to prevent the app from quitting programmatically, as iOS device users tend to interpret this as crashing.

Z. A License Agreement or Disclaimer is displayed when it is necessary

Include an end-user license agreement (EULA) or a disclaimer when necessary, as it will be displayed by the App Store so that users can read it before they acquire the app.

IV. USABILITY EVALUATION METHODOLOGY AND PRELIMINARY RESULTS

This section presents an example of the use of the expert-based evaluation framework for iOS app usability. An iPhone 4 running on iOS version 6.1.3 is used for these experiments which were designed to evaluate 11 iOS apps selected from Apple's App Store with the following characteristics:

- Type: utility or business
- Price: 0
- An available recorded version
- An available back-up for further research

Criteria are proposed in Table I to implement the framework for evaluating iOS app usability. These criteria are used to evaluate the usability of the 11 selected apps.

The criteria are related to 6 different heuristics (from A to F), and were entered on an Excel spreadsheet. The answers are numbered from 1 to 5 in ordinal scale for each criteria. Also, inapplicable questions were not asked, and were marked as N/A (not applicable). Two different evaluators, co-authors of this paper, evaluated the criteria for each app, and the results were recorded. Table II presents the results of evaluations for each app by each evaluator by counting the number of answers for each choice. Answers are choices from 1 to 5 for each criteria, with choices 4 and 5 considered as an indication of better overall app usability.

Furthermore, the percentage of chosen 4 and 5, over the number of answers eliminating the neutral choice (3), is calculated for each evaluator and the average of the two evaluators are presented in table III.

In addition to the expert-based evaluations, user ratings provided by Apple's App Store customers, as well as rating counts, are presented in Table III. User ratings computed as percentage of given 4 and 5s over the sum of 1, 2, 4 and 5 choices. Hence the neutral choice (3) is eliminated for user rating calculation also.

TABLE I. GUIDELINE EVALUATION CRITERIA

	Guidelines
A	**User control and freedom**
1	Application has marked exit
2	Application support undo and redo
3	User can control interaction
4	User knows where he is in, where he can go and how he get there in application
5	User understands how many steps he will go to reach his goal and his current position
B	**Error correction**
1	When something went wrong, application tells the user: what happened, why it happened and how to fix it
2	Required fields are made obvious with visual indications
3	Back button/gesture turns to previous view and the data is not lost
C	**Accommodation**
1	Application speaks user's language
2	Relevant metaphors are used when needed
3	Interface is suitable for the user's task and skill level
D	**Linguistic clarity**
1	There is no spelling or grammar mistakes
2	Abbreviations and acronyms are not used if they are not straight forward
E	**Simplicity**
1	Minimalist or view what is relevant for the need
2	Different colors are used for different purposes
3	Size of graphics is considered for response time impact
4	Used of colors is limited (3-4)
5	Application's purpose is understandable at the first sight
F	**Aesthetics**
1	Similarity
2	Continuation
3	Closure
4	Proximity
5	Figure and ground

TABLE II. EXPERTS EVALUATION RESULTS OF THE SELECTED APPS

	Evaluator 1						**Evaluator 2**					
App	**1**	**2**	**3**	**4**	**5**	**N/A**	**1**	**2**	**3**	**4**	**5**	**N/A**
1	7	4	0	2	1	9	1	6	8	5	1	2
2	0	0	2	12	7	2	0	0	3	8	10	2
3	0	0	2	10	9	2	0	0	2	9	10	2
4	2	9	4	5	2	1	1	2	12	3	3	2
5	0	2	2	12	6	1	2	7	6	4	2	2
6	0	1	2	14	5	1	0	2	2	10	8	1
7	0	1	4	7	7	4	0	3	6	9	3	2
8	0	0	1	6	10	6	0	0	4	16	0	3
9	3	0	3	5	9	3	1	1	5	3	9	4
10	0	1	4	10	5	3	0	2	7	8	4	2
11	0	0	1	10	11	1	0	1	4	9	8	1

Correlations were calculated to analyze the relationship between the two experts evaluators, and between the experts average evaluations and the App Store user ratings.

Table IV presents these correlations, which are as follows: the correlation between the two evaluators for all results is 0.67. When app 5 is removed, the correlation is 0.94. This shows that the two evaluators' evaluations correlate well and even better, if app 5 is removed.

Table IV also shows the correlation between the average expert evaluations and App Store user ratings, which is very low at 0.26. When apps with fewer than 100 rating counts are

TABLE III. OVERALL RESULT OF EVALUATION

App	Evaluator 1: % of 4 & 5	Evaluator 2: % of 4 & 5	Average of Evaluator1 & Evaluator2	User Average Rating	Reviews Count
1	0.21	0.46	0.34	0.30	9
2	1.00	1.00	1.00	0.48	29
3	1.00	1.00	1.00	0.75	5
4	0.39	0.68	0.53	1.00	13
5	0.90	0.40	0.65	0.42	4911
6	0.95	0.90	0.92	0.53	10
7	0.93	0.80	0.87	0.29	281
8	1.00	1.00	1.00	0.86	5621
9	0.82	0.86	0.84	0.86	140
10	0.94	0.86	0.90	0.58	12
11	1.00	0.94	0.97	0.98	323

TABLE IV. CORRELATIONS BETWEEN EXPERT EVALUATIONS AND USER RATINGS

Correlation Element	Correlation Value
Between evaluators	0.67
Between evaluators (without app 5)	0.94
Between evaluation averages and App Store ratings	0.26
Between evaluation averages and App Store ratings for apps with more than 100 reviews	0.66

removed (6 apps), the correlation is 0.66 which is significantly better. This result shows that review counts below 100 are not reliable enough.

In this study, in which only 11 apps were evaluated, also a very small subset of usability criteria was used and answered. With this limited number of questions and applications we have 0.66 as correlation which is acceptable for this study. With evaluating more applications and more criteria it is probable that we will have better correlations.

We can see from Tables 3 and 4 that:

- The average value of an evaluation varies from 0.34 to 1 for the 11 apps.
- The evaluations of the two experts are consistent, disregarding A5 (app 5).
- In general, evaluator 1 rated all the apps more highly than did evaluator 2.
- Correlation between Apple App Store given user ratings and expert-based usability evaluation averages is acceptable in general for applications with review counts over 100.

V. CONCLUSION AND FUTURE WORKS

This study has presented a framework for evaluating iOS app usability. Figures 1 and 2 present guidelines found in the literature, while Figure 3 presents industry-based usability guidelines proposed by Apple for iOS apps. These are the guidelines to consider in app design.

All the guidelines and heuristics discussed in this study have been converted to usability criteria, based on the evaluation criteria proposed in sections II and III. In section IV, criteria were selected for quantifying the usability of 11 different apps by two different expert evaluators. The preliminary results show that the selected subset of criteria for evaluating the usability of applications with user reviews over 100, generated acceptable results. Perhaps, expert-based evaluations could be combined with user-based evaluations, due to the importance of user-based evaluations, to have more accurate results. Furthermore, the relationship between usability and App Store user ratings should be examined more deeply with answering all proposed criteria for each application as well as combining those results with user-based evaluations.

We suggest addressing the following issues in future work:

- improvement of the formulation of the expert-based evaluation criteria;
- inclusion of all the formulated expert-based evaluation criteria;
- evaluation of more apps and including more evaluators to answer the questions;
- employment of user-based evaluation questionnaires; and
- application of feature subset selection to eliminate meaningless criteria.

REFERENCES

[1] Fatih Nayebi, Jean-Marc Desharnais, Alain Abran, "The State of the Art of Mobile Application Usability Evaluation", 25th IEEE Canadian Conference on Electrical and Computer Engineering (IEEE CD: 978-1-4673-6/12) Montreal, April 29-May 2, 2012.

[2] Apple iOS Human Interface Guidelines, 2012-12-17 updated version: http://developer.apple.com/library/ios/#documentation/UserExperience /Conceptual/MobileHIG/

[3] Jacob Nielsen, "Usability Engineering". San Diego: Academic Press. pp. 115–148. ISBN 0-12-518406-9, 1994.

[4] Jill Gerhardt-Powals, "Cognitive Engineering Principles for Enhancing Human-Computer Performance". International Journal of Human-Computer Interaction, 8(2), 189-21, 1996.

[5] Ben Shneiderman, Catherine Plaisant, Maxine Cohen, Steven Jacobs. "Designing the User Interface: Strategies for Effective Human-Computer Interaction, 5th edition". Addison-Wesley. ISBN 0-321-26978-0, 2010.

[6] Susan Weinschenk and Dean T. Barker, "Designing effective speech interfaces", Wiley computer publishing ISBN 9780471375456, 2000.

[7] William Edmund Hick, "On the rate of gain of information. Quarterly Journal of Experimental Psychology", 4:11-26, 1952.

[8] Ray Hyman, "Stimulus information as a determinant of reaction time". Journal of Experimental Psychology, 45:188-196, 1953.

[9] Paul M. Fitts, "The information capacity of the human motor system in controlling the amplitude of movement". Journal of Experimental Psychology, volume 47, number 6, June 1954, pp. 381–391. (Reprinted in Journal of Experimental Psychology: General, 121(3):262–269, 1992).

[10] Fibonacci Number: http://mathworld.wolfram.com/FibonacciNumber.html

[11] Alan Baker, "Simplicity", Stanford Encyclopaedia of Philosophy, California: Stanford University, ISSN 1095-5054, retrieved 25 July 2012.

[12] Richard Koch, "The 80/20 Principle: The Secret of Achieving More with Less", London: Nicholas Brealey Publishing, 2001.

[13] Sandra Meech, "Contemporary Quilts: Design, Surface and Stitch". Sterling Publishing. ISBN 0-7134-8987-1, 2007.

[14] Mario Livio, "The Golden Ratio: The Story of Phi, The World's Most Astonishing Number". New York: Broadway Books. ISBN 0-7679-0815-5, 2003.

[15] Christopher D. Wickens, John Lee, Yili D. Liu, Sallie Gordon-Becker, "An Introduction to Human Factors Engineering". Pearson. ISBN 0131837362, 2003.

APPENDIX III

SYSTEMATIC REVIEW RESULTS

This Appendix presents the 38 studies qualified as final studies to take place in systematic literature review on iOS application usability evaluation.

[1] Passani, Luca. "Building usable wireless applications for mobile phones." In Human Computer Interaction with Mobile Devices, pp. 9-20. Springer Berlin Heidelberg, 2002.

[2] Wright, Tim, Pak Yoong, James Noble, Roger Cliffe, Rashina Hoda, Donald Gordon, and Chris Andreae. "Usability methods and mobile devices: an evaluation of MoFax." In Proceedings of the 4th international conference on Mobile and ubiquitous multimedia, pp. 26-33. ACM, 2005.

[3] Zhang, Dongsong, and Boonlit Adipat. "Challenges, methodologies, and issues in the usability testing of mobile applications." International Journal of Human-Computer Interaction 18, no. 3 (2005): 293-308.

[4] Juhasz, Z., Arato, A., Bognar, G., Buday, L., Eberhardt, G., Markus, N., ... & Vaspori, T. (2006). Usability evaluation of the MOST mobile assistant (slattalker). In Computers Helping People with Special Needs (pp. 1055-1062). Springer Berlin Heidelberg.

[5] Coursaris, Constantinos, and Dan Kim. "A qualitative review of empirical mobile usability studies." AMCIS 2006 Proceedings (2006): 352.

[6] Al-Qaimari, Ghassan, and Shane Fernando. "Location-aware applications: evaluating the ease of use and ease of learning." In Proceedings of the 2006 international conference on Wireless communications and mobile computing, pp. 1283-1288. ACM, 2006.

[7] Ryu, Young Sam, and Tonya L. Smith-Jackson. "Reliability and validity of the mobile phone usability questionnaire (MPUQ)." (2006).

[8] Qiu, Yuan Fu, Yoon Ping Chui, and Martin G. Helander. "Usability analysis of mobile phone camera software systems." In Cybernetics and Intelligent Systems, 2006 IEEE Conference on, pp. 1-6. IEEE, 2006.

[9] Pousttchi, Key, and Bettina Thurnher. "Understanding effects and determinants of mobile support tools: A usability-centered field study on IT service technicians." In Mobile Business, 2006. ICMB'06. International Conference on, pp. 10-10. IEEE, 2006.

[10] Ji, Yong Gu, Jun Ho Park, Cheol Lee, and Myung Hwan Yun. "A usability checklist for the usability evaluation of mobile phone user interface." International Journal of Human-Computer Interaction 20, no. 3 (2006): 207-231.

[11] Schultz, David. "10 usability tips & tricks for testing mobile applications." interactions 13, no. 6 (2006): 14-15.

[12] Tesoriero, Ricardo, María Dolores Lozano, José A. Gallud, and Victor M. Ruiz Penichet. "Evaluating the Users' Experience of a PDA-based Software Applied in Art Museums." In WEBIST (2), pp. 351-358. 2007.

[13] Jambon, Francis, Caroline Golanski, and P-J. Pommier. "Meta-evaluation of a context-aware mobile device usability." In Mobile Ubiquitous Computing, Systems, Services and Technologies, 2007. UBICOMM'07. International Conference on, pp. 21-26. IEEE, 2007.

[14] Bahn, S., C. Lee, J. H. Jo, W. Y. Suh, J. Song, and M. H. Yun. "Incorporating user acceptance into usability evaluation scheme for the user interface of mobile services." In Industrial Engineering and Engineering Management, 2007 IEEE International Conference on, pp. 492-496. IEEE, 2007.

[15] Das, Bhagaban, and Sangeeta Mohanty. "Service Usability and Users' Satisfaction in India: An Exploratory Study on Mobile Phone Users." ICFAI Journal of Services Marketing 5, no. 4 (2007).

[16] Concejero, Pedro, J. Patrocinio, and D. Merino. "Usability evaluation of mobile services." In proc. ICIN, vol. 8. 2008.

[17] Vuolle, Maiju, Mari Tiainen, Titti Kallio, Teija Vainio, Minna Kulju, and Heli Wigelius. "Developing a questionnaire for measuring mobile business service experience." In Proceedings of the 10th international conference on Human computer interaction with mobile devices and services, pp. 53-62. ACM, 2008.

[18] Hummel, Karin A., Andrea Hess, and Thomas Grill. "Environmental context sensing for usability evaluation in mobile HCI by means of small wireless sensor networks." In Proceedings of the 6th International Conference on Advances in Mobile Computing and Multimedia, pp. 302-306. ACM, 2008.

[19] Fetaji, Majlinda, Zamir Dika, and Bekim Fetaji. "Usability testing and evaluation of a mobile software solution: a case study." In Information Technology Interfaces, 2008. ITI 2008. 30th International Conference on, pp. 501-506. IEEE, 2008.

[20] Gafni, Ruti. "Usability issues in mobile-wireless information systems." Issues in Informing Science and Information Technology 6 (2009): 755-769.

[21] Anuar, Nor Badrul, Lai Ngan Kuen, Omar Zakaria, Abdullah Gani, and Ainuddin Wahid Abdul Wahab. "Usability and performance of secure mobile messaging: M-PKI." WSEAS Transactions on Information Science and Applications 6, no. 2 (2009): 179-189.

[22] Balagtas-Fernandez, Florence, and Heinrich Hussmann. "A methodology and framework to simplify usability analysis of mobile applications." In Automated Software Engineering, 2009. ASE'09. 24th IEEE/ACM International Conference on, pp. 520-524. IEEE, 2009.

[23] Heo, Jeongyun, Dong-Han Ham, Sanghyun Park, Chiwon Song, and Wan Chul Yoon. "A framework for evaluating the usability of mobile phones based on multi-level, hierarchical model of usability factors." Interacting with Computers 21, no. 4 (2009): 263-275.

[24] Hussain, Azham, and Maria Kutar. "Usability metric framework for mobile phone application." PGNet, ISBN (2009): 978-1.

[25] Pham, Tan Phat, Khasfariyati Razikin, Dion Hoe-Lian Goh, Thi Nhu Quynh Kim, Huynh Nhu Hop Quach, Yin-Leng Theng, Alton YK Chua, and Ee-Peng Lim. "Investigating the usability of a mobile location-based annotation system." In Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia, pp. 313-320. ACM, 2010.

[26] Maly, Ivo, Zdenek Mikovec, and Jan Vystrcil. "Interactive analytical tool for usability analysis of mobile indoor navigation application." In Human System Interactions (HSI), 2010 3rd Conference on, pp. 259-266. IEEE, 2010.

[27] Lee, Young Seok, Santosh Basapur, Harry Zhang, Claudia Guerrero, and Noel Massey. "Usability evaluation of beep-to-the-box." In Proceedings of the 12th international conference on Human computer interaction with mobile devices and services, pp. 345-348. ACM, 2010.

[28] Moritz, Franka, and Christoph Meinel. "Mobile Web Usability Evaluation-Combining the Modified Think Aloud Method with the Testing of Emotional, Cognitive and Conative Aspects of the Usage of a Web Application." In Computer and Information Science (ICIS), 2010 IEEE/ACIS 9th International Conference on, pp. 367-372. IEEE, 2010.

[29] Chua, Alton Yeow-Kuan, Dion H. Goh, Chei-Sian Lee, and Keng-Tiong Tan. "Mobile alternate reality gaming engine: A usability evaluation." In Information Technology: New Generations (ITNG), 2010 Seventh International Conference on, pp. 540-545. IEEE, 2010.

[30] Billi, Marco, Laura Burzagli, Tiziana Catarci, Giuseppe Santucci, Enrico Bertini, Francesco Gabbanini, and Enrico Palchetti. "A unified methodology for the evaluation of accessibility and usability of mobile applications." Universal Access in the Information Society 9, no. 4 (2010): 337-356.

[31] Kunjachan, Mary Ann Chiramattel. "Evaluation of Usability on Mobile User Interface." University of Washington, Bothell (2011).

[32] Biel, Bettina, Thomas Grill, and Volker Gruhn. "Exploring the benefits of the combination of a software architecture analysis and a usability evaluation of a mobile application." Journal of Systems and Software 83, no. 11 (2010): 2031-2044.

[33] Hegarty, Ronan, and Judith Wusteman. "Evaluating EBSCO host Mobile." Library Hi Tech 29, no. 2 (2011): 320-333.

[34] Coursaris, Constantinos K., and Dan J. Kim. "A meta-analytical review of empirical mobile usability studies." Journal of usability studies 6, no. 3 (2011): 117-171.

[35] Coronel, Nazir O. Molina, J. Ortiz Hernandez, S. Saenz Sanchez, and Juan G. Gonzalez Serna. "Integration of Usability Into the Development of Mobile Computing Environments Applied to Contextual Location Based Services (LBS)." In Electronics, Robotics and Automotive Mechanics Conference (CERMA), 2011 IEEE, pp. 409-414. IEEE, 2011.

[36] Jeong, Wooseob, and Hyejung Han. "Usability study on mobile Web newspaper sites." Proceedings of the American Society for Information Science and Technology 48, no. 1 (2011): 1-4.

[37] Hashim, Ahmad Sobri, Wan Fatimah Wan Ahmad, and Rohiza Ahmad. "Mobile learning course content application as a revision tool: The effectiveness and usability." In Pattern Analysis and Intelligent Robotics (ICPAIR), 2011 International Conference on, vol. 2, pp. 184-187. IEEE, 2011.

[38] Ji, Yong Gu, Jun Ho Park, Cheol Lee, and Myung Hwan Yun. "A usability checklist for the usability evaluation of mobile phone user interface." International Journal of Human-Computer Interaction 20, no. 3 (2006): 207-231.

APPENDIX IV

LIST OF THE 99 SELECTED iOS APPLICATIONS

This Appendix presents the list of 99 iOS applications that were selected for experimentation.

No	APPs	Category	Version	1 Star	2 Stars	3 Stars	4 Stars	5 Stars
1	Excel notepad	Productivity	1.1	7	1	0	2	1
2	Opera Mini	Productivity	7.0.5	11	9	11	8	16
3	Skitch	Productivity	2.0.5	44	9	14	32	96
4	30/30	Productivity	2.1	0	2	0	3	18
5	iDoodle2 lite	Productivity	1.6	1246	939	1114	554	1054
6	SimpleNote	Productivity	3.3.2	133	83	119	83	166
7	SpeedTest X HD	Utility	5.4.5	103	58	48	32	56
8	Planets	Utility	3.4	994	786	1218	817	1819
9	Calculette	Utility	1.3	50	12	39	102	105
10	CinemaClock	Utility	1.2	19	7	9	12	24
11	Any-Do	Productivity	1.8.1	30	16	53	356	1222
12	AnyList	Productivity	2.0.1	6	3	4	15	99
13	Pages jaunes	Productivity	3.8.0	2411	779	1183	1313	3844
14	Box	Business	2.8.4	154	94	152	148	225
15	Camcard Free	Productivity	4.1.0.0	55	21	60	270	963
16	Catch Notes	Productivity	5.2.6	25	9	8	30	86
17	Decibel Meter Pro	Utility	2.0.5	88	37	60	211	511
18	Flightradar24	Productivity	4.7.1	273	88	51	36	79
19	GoDocs	Utility	4.2	890	222	165	155	426
20	Inkflow: Think Visually	Productivity	2.5	64	54	74	194	1076
21	LogMein	Business	3.3.2119	670	378	499	2502	17522
22	Magic Plan	Utility	3.1.1	125	88	90	176	480
23	My Eyes Only Classic	Business	1.2.1	162	87	103	247	798
24	N+otes	Productivity	4.8.9	144	61	47	71	180
25	Scanner Pro	Business	4.5	498	210	273	946	5210
26	SignEasy	Business	4.2.6	248	129	261	1309	4944
27	SkyDrive	Productivity	3.0.1	443	268	373	409	1204
28	Speedtest mobile	Utility	3.1	8951	4827	9262	10772	23174
29	Transit	Utility	2.0.3	105	77	54	98	484
30	Cardiographe	Health	2.4.3	31	4	7	18	35
31	Universel Unit Converter	Utility	2.3	131	52	78	276	837
32	My measures	Productivity	4.06	69	13	14	35	83
33	Receipts by Wave	Finance	1.2	181	40	67	174	914
34	Istudiez Pro	Productivity	1.6.6	13	6	3	11	34

No	APPs	Category	Version	1 Star	2 Stars	3 Stars	4 Stars	5 Stars
35	Mes Tim	Life style	4	934	458	714	1212	2993
36	Egg Timer	Life style	1	4	0	1	1	10
37	Wikipedia Mobile	References	3.3.1	7	6	0	1	28
38	Transit directions	Utility	1.6.3	13027	8240	9160	6588	16299
39	Solar Weather	Weather	1.2	127	34	51	146	429
40	Haze	Weather	1.03	16	21	31	131	358
41	Canada Post	Business	5.1.7	1119	368	180	147	362
42	Voice translator	Business	3.6.0	434	144	26	15	45
43	Language Translator (Piet Jonas)	Business	1.1.1	648	208	471	490	1156
44	Speak and Translate	Business	1.0	8	6	1	0	0
45	Roambi Analytics	Business	7.2.0	47	36	32	31	69
46	MicroStrategy Mobile	Business	9.4.11	16	5	3	5	20
47	Keynote	Business	2.1	432	129	103	76	93
48	Page	Business	2.1	1095	482	320	222	356
49	Numbers	Business	2.1	278	122	90	89	155
50	Concur Mobile	Business	9.10.0	572	256	31	28	47
51	Things	Business	2.2.5	689	365	187	99	148
52	Skygrid	Business	2.6	510	314	109	77	104
53	Delivery Status Touch	Business	5.0.1	233	65	26	12	21
54	Shoeboxed	Business	4.5.2	30	14	16	13	53
55	Invoice2go Plus	Business	7.7.1	126	76	12	9	31
56	Calvetica Calendar	Productivity	5.2.1	134	46	24	19	44
57	Adobe Ideas	Entertainment	2.7.2	164	104	102	77	128
58	ScatterBrain (Tomasello)	Productivity	1.3.3	182	43	9	3	4
59	Weave	Productivity	1.23	94	54	10	4	3
60	Do it (Tomorrow)	Productivity	2.0.1	387	232	95	65	117
61	Awesome Note	Productivity	7.1	788	353	165	118	197
62	Cookspiration	F & B	1.0	28	5	4	0	2
63	Wine Spectator WineRatings+	F & B	4.0.1	70	130	72	24	48
64	Epicurious Recipes & Shopping List	F & B	4.0.2	838	564	694	574	834
65	How to cook everything Culinate	F & B	1.9.12	460	115	22	13	13
66	Hello Vino Wine Assistant	F & B	3.7.1	70	65	115	102	167

No	APPs	Category	Version	1 Star	2 Stars	3 Stars	4 Stars	5 Stars
67	Icookbook - thousands of ...	F & B	3.4.1	71	36	11	7	20
68	Must have recipes Better Homes and Gardens	F & B	3.1.3	84	19	8	7	18
69	Grocery IQ	F & B	2.7.1	244	267	230	212	334
70	Evernote Food	F & B	2.3.2	107	55	17	5	12
71	Julian Michaels Slim-Down	Health	5.1.0	401	141	80	55	186
72	Runtastic pro GPS course	Health	3.1.3	1193	140	26	23	28
73	RunKeeper - GPS courir Marcher	Health	4.3.1	2203	476	119	71	160
74	FitStar: Tony Gonzalez	Health	2.3.0	76	15	1	2	3
75	Yoga Studio	Health	2.1.2	739	77	7	2	6
76	Pocket yoga	Health	3.2.0	252	94	20	7	12
77	MotionX 24/7: Sleep Cycle Alarm	Health	8.1	82	25	5	1	4
78	TuneIn Radio	Music	5.1	10268	1968	330	155	239
79	Shazam	Music	7.4.1	23367	6193	5184	3491	10785
80	Garage Band	Music	2.0.1	1421	341	165	128	193
81	Band of the day	Music	3.1.4	339	36	10	8	10
82	Figure Propellerhead	Music	1.5.3	260	94	38	14	15
83	Magic Piano	Music	6.1.1	2152	812	332	200	258
84	RDS GO	Sport	3.0.8	1557	316	152	63	137
85	Juxtaposer	Photo	3.0.2	101	23	12	9	9
86	Filpagram	Photo	2.9.7	1811	285	56	20	30
87	Over (In App Purchases - Potluck)	Photo	2.2.2	229	92	37	22	41
88	Rookie - photo editor	Photo	1.0.1	99	38	14	2	13
89	Repix	Photo	1.5.3	62	24	11	8	16
90	Google Translate	References	2.1.1	1028	279	211	177	531
91	Bing Search - trends...	References	4.4	99	44	73	62	138
92	Barefoot World Atlas	References	3.0.3	27	11	12	18	28
93	Merriam-Webster	References	3.0.1	78	15	2	3	6
94	ShopSavvy Classic Bar Code	Utility	8.7.0	310	162	249	286	1043
95	Ancestry	References	5.1.1	895	526	185	154	516

No	APPs	Category	Version	1 Star	2 Stars	3 Stars	4 Stars	5 Stars
96	National Geographic World Atlas	References	3.4	257	163	59	50	76
97	NHL GameCenter	Sports	4.0121	585	260	239	278	1298
98	Yahoo Sports	Sports	5.0.3	1931	1003	836	467	639
99	TSN Go	Sports	1.1.6	5047	2017	1745	1309	3109

APPENDIX V

ISO 9241 - ERGONOMIC REQUIREMENTS FOR OFFICE WORK WITH VISUAL DISPLAY TERMINALS

This Appendix presents a summary of ISO 9241 - Ergonomic requirements for office work with visual display terminals standard.

1. ISO 9241 - 10: Dialogue principles (1996)

This part deals with general ergonomic principles relating to the design of dialogues connecting humans and information systems. Principles include suitability for the task, suitability for learning, suitability for individualization, conformity with user expectations, self-descriptiveness, controllability, and error tolerance.

2. ISO 9241 - 11: Guidance on Usability (1998)

This part presents the definition of usability in subsequent related ergonomic standards and defines usability as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use".

ISO 9241-11 explains how to distinguish the required information when designating or assessing usability concerning measures of user performance and satisfaction. Additionally, this part guides on how to specify the context of use of the application and the measures of usability in an unambiguous fashion. It includes an explanation of how the usability of an application can be designated and assessed as part of a quality system, for instance, one that complies to ISO 9001. It also describes how measures of user performance and satisfaction can be used to measure how any component of a working system affects the quality of the whole work system in use.

3. ISO 9241 - 12: Presentation of information (1998)

This part contains recommendations for presenting and representing information on visual displays. It involves guidance on ways of expressing complex information using alphanumeric and graphical/symbolic codes, view layouts, and design.

4. ISO 9241 - 13: User guidance (1998)

This part suggests the design and evaluation of user guidance characteristics of software UI covering prompts, feedback, status, help and error management.

5. ISO 9241 - 14: Menu dialogues (1997)

This part provides suggestions on the design of menu dialogues. The suggestions incorporate structure, navigation, option selection and execution, and presentation.

6. ISO 9241 - 15: Command dialogues (1997)

This part provides suggestions on the design of command languages used in user-computer dialogues. The suggestions include command language structure and syntax, command representations, input and output considerations, feedback and help.

7. ISO 9241 - 16: Direct manipulation dialogues (1999)

This part provides recommendations for the direct manipulation dialogues' ergonomic design considering the design of metaphors, objects and attributes. It includes directly manipulated Graphical UIs aspects, and not embraced by other parts of ISO 9241.

8. ISO 9241 - 17: Form filling dialogues (1998)

This part provides suggestions on the form filling dialogues' ergonomic design. The suggestions cover form structure and navigation as well as the input and output considerations.

9. ISO 9241 - 151: Guidance on World Wide Web user interfaces

This part provides suggestions and guidelines for the user-centered design of the web UIs.

10. ISO 9241 - 210: Human-Centered design for interactive systems

This standard provides guidance on human-centered design activities throughout the development life cycle of interactive computer-based systems. It is a tool to be used in design processes management and provides guidance on information sources and standards of human-centered approach.

Human-Centered design is expressed as a multi-disciplinary exercise, consolidating human factors and ergonomics to enhance effectiveness and efficiency, improve human working circumstances, and counteracting possible adverse effects of use on human health, safety and performance.

Any user-centered design exercise may plan and undertake four essential activities to incorporate usability requirements into the development process. Activities are as the following:

- Understand and specify the context of use.

  This may address the following essential aspects:

  - The characteristics of the intended users
  - The tasks the users will perform
  - The usage environment

- Specify the user and organizational requirements.

  This looks at the user and organizational requirements in relation to the context of use description, and it may:

  - Distinguish the range of related users and other stakeholders in the design.

  - Provide a clear statement of the human-centered design objectives.
  - Set proper priorities for the varied requirements.
  - Provide well-defined measures to evaluate emerging designs.
  - Be confirmed by the users and other stakeholders.
  - Include statutory or legislative requirements.
  - Be adequately documented

- Produce design solutions
  - Develop high-level design schemes with multi-disciplinary inputs.
  - Develop low-level and concrete design schemes using simulations and mock-ups.
  - Present the designs to users and other stakeholders and enable them to perform or simulate tasks.
  - Iterate the process until meeting the design objectives.

- Evaluate designs against requirements.

  This is an essential step that evaluates whether user and organizational objectives have been met and provides feedback to improve the design. There is a variety of evaluation methods, differing in their formality and user involvement - the best methods will depend on the essence of the application being developed, finances, and time constraints.

APPENDIX VI

OTHER GUIDELINES TO CONSIDER DURING APPLICATION DESIGN

1. Hick-Hyman's law

Hick (1952) and Hyman (1953) state that the time it takes to perform a selection grows with every additional option available. This implies that the more options a user has when using a mobile application, the more complicated it will be to use. Therefore, underlines the importance of simplicity. Given n equally probable choices, the average response time T required to choose among them is almost:

$$T = b * \log_2(n+1) \tag{A VI-1}$$

where b is a constant, that can be learned empirically by fitting a line to the observation dataset. This logarithmic operation expresses depth of the choice tree hierarchy. $Log2$ denotes that a binary search is conducted. The law can be inferred to the case of choices with unlike probabilities p_i occurring, with:

$$T = bH \tag{A VI-2}$$

where H is the information-theoretic entropy of the decision, explained as

$$H = \sum_{i}^{n} p_i \log_2(1/p_i + 1) \tag{A VI-3}$$

where p_i refers to the probability of the i^{th} alternative yielding the information-theoretic entropy. This law implies that designers should minimize the number of choices a user possesses. Removing unnecessary views, navigations, buttons, or selections will make designs more useful.

2. Fitt's law

According to Fitt, "the time required to move to a target is a function of the target size and distance to the target." Fitts (1954). Designers can apply this to app design by looking at the hit

area of the objects used, meaning that the larger the tappable area of the navigational elements, the easier it will be to tap.

Fitt's guideline has been formulated mathematically in several different ways. A common form is the Shannon formulation (proposed by MacKenzie, and named for its resemblance to the Shannon-Hartley theorem) for movement along a single dimension:

$$T = a + b * \log_2(1 + D/W) \qquad \text{(A VI-4)}$$

where:

- T is the average time taken to complete the movement. (Traditionally, researchers have used the symbol MT for this, which stands for movement time.)
- a represents the start/stop time of the device (intercept)
- b stands for the inherent speed of the device (slope).

  Constants a and b can be determined experimentally by fitting a straight line to the measured data.
- D is the distance from the starting point to the center of the target. (Traditionally, researchers have used the symbol A for this, which stands for the amplitude of the movement.)
- W is the width of the target measured along the axis of motion. It can also be thought of as the error tolerance allowed in the final position of the final point of the motion must fall within $W/2$ of the target's center.

From the equation, we can see that a speed-accuracy tradeoff is associated with pointing, that is, targets that are smaller and/or further away require more time to acquire.

3. Fibonacci sequence

The Fibonnaci sequence is a series of numbers in which each number is the sum of the preceding two. For example, starting with 1, the sequence is the following:

$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, etc...$

The Fibonacci sequence is one of the most significant patterns in both mathematics and design, and this has been recognized in many classical works. It is commonly found in nature, and, like the golden ratio (see below), it is often used to create visual patterns, shapes, and organic figures, to build grids, and to dictate sizing and ratios. Moreover, patterns based on this sequence are intrinsically esthetic, which is an added advantage to its use in design composition.

4. Occam's razor

According to Baker (2012) Occam's razor suggests "the simplest solution is almost always the best". With the flexibility and power of the mobile device and its design tools, it is easy to be carried away in the design process. The result could be a very complicated app or design with many functionalities and information, but difficult to build, use, and maintain. The app may promise more, but achieve less.

This is commonly an issue when a company feels the need to put everything they possibly can into a mobile app, in the belief that someone will want the information. What such a company fails to consider is that the overwhelming majority of users will access only about 20% of the content.

Being ruthless about the value that a view or piece of content produces and removing any content that is unnecessary will result in significantly stronger and more effective designs.

This guideline also implies “a design is not finished when there is nothing more to add, but when there is nothing left to take away.” Design simplicity is elegant, sophisticated, and much more effective than the elaborate decorative style that is so prevalent in mobile applications.

5. Pareto guideline

According to the Koch (2001) Pareto guideline, implies a high percentage of users will perform a small proportion of actions. This means that most users will go to a limited figure of views. In the case of mobile applications, most users will also perform a limited number of tasks.

Based on this guideline, a designer can identify what actions most users will perform (from analytics, research, and interviews), and then put greater emphasis on those tasks and activities to make the application easier to use. Sometimes this can involve the inclusion of a new navigation feature, or modification of the homepage to make finding and accomplishing the tasks easier.

It can also involve paring down and removing content and features from a mobile app. If users are not accessing or using the information, then the usability of the application can be improved by eliminating it. This leads us back to Hick’s Law and Occam’s Razor.

6. Rule of thirds

The Meech (2007) explains rule of thirds as a method of composing elements to make them visually pleasing, as well as to identify ways in which the user’s eyes will scan the view. Photographers have been using this heuristic for years to create more visually interesting compositions.

The rule of thirds operates by breaking up a design into thirds, both vertically and horizontally, and building a grid of intersecting lines. According to this rule, a viewer is more likely to be drawn to the intersections than to the lines or the spaces between them. Good rules of thumb

are to place elements along the lines and at the intersections, and to avoid placing anything in the dead center of the composition or having a horizon dividing the composition in half.

Placing elements so that they take up one-third or two-thirds of the space will be more visually pleasing to most viewers.

7. Golden ratio

The golden ratio is often confused with the rule of thirds. In fact, this ratio looks at what proportions are naturally the most visually appealing. It has been used in the design, architecture, and engineering for hundreds of years. According to Livio (2003) the golden ratio is calculated using the elements of a shape, such as height to width, and turns out to be approximately 0.618.

When applied to rectangles, the golden ratio can be used by a designer to break the shape down into smaller renderings, creating a natural spiral pattern. This pattern can be seen in nature by examining sea shells.

8. Proximity

According to Wikipedia (b) proximity, suggests that elements close to one another will appear related, a fact that is often overlooked. What this means is that a designer must be aware of how much space he is placing between the elements of a design. If the elements of a series are too close together, the user will assume that this has been done intentionally and that the elements are related. This is often an issue with mobile applications, where buttons or controls are grouped together, and yet have unrelated functionality. The result is confusion on the part of the user, who is trying to understand the application.

9. Feedback

Feedback is a concept that industrial designers have mastered for decades. Feedback is giving a user clear indication that something has happened, is happening or could happen.

Since users interact with mobile applications, designers need to be aware of providing adequate feedback.

10. Mental Models

The Mental Model guideline affirms that it is significantly easier for users to understand and learn a new thing if they can compare it with something they already understand.

Designers can use this concept to make designs easier to use and more effective visually. There are circumstances where it would be effective to model designs off of real world situations or objects. Users can learn, understand and extract meaning from certain types of designs because they can associate it with their understanding of the objects in real life.

11. Cognitive biases and user experience

According to Wikipedia (a), people develop beliefs regarding how things should appear, how things should act, and what things should be named. These cognitive biases constitute a filter between what exists, and what one perceive to be true.

The field of user experience (UX) design attempts to recognize a user's cognitive biases or assumptions and deliberates design decisions that make use of those biases.

If a UX designer does not identify and incorporate users' cognitive biases into his work, it stands to be misrepresented, and application goals stand to be unachieved.

The followings sub-sections present examples of cognitive biases of application users.

11.1 I know where it should be located

Users tend to have a preconceived understanding of where particular UI views and controls are likely to be placed.

Consideration: Placement of key/common UI controls and views that are essential to completing critical user stories can be plied with attention.

11.2 I know how it should look

Users develop preconceived notions regarding how, many UI controls should appear.

Consideration: Visual appearance of key UI controls and groupings of content is relevant to the user experience. Analogies to real word objects should be adhered to but should not be taken literally. Principles of Gestalt psychology could be applied in order to compose logical visual hierarchies and content groups.

11.2.1 Gestalt perception principles

Gestalt is a psychology term meaning, "unified whole". It introduces the theories of visual perception. These theories try to describe how users tend to organize UI elements into groups or unified wholes when specific principles are applied. These principles are:

11.2.1.1 Similarity

Similarity happens when objects look similar to one another. People often perceive them as a group or pattern.

11.2.1.2 Continuation

Continuation happens when the eye is compelled to move through one object and continue to another object.

11.2.1.3 Closure

Closure happens when an object is unfinished, or a space is not completely surrounded. If enough of the shape is shown, users perceive the whole by filling in the missing information.

11.2.1.4 Proximity

Proximity happens when UI controls are placed close together. They tend to be perceived as a group.

11.2.1.5 Figure and Ground

The eye distinguishes an object from its enclosing area. A form, silhouette, or shape is naturally perceived as a figure while the surrounding area is viewed as ground. According figure and ground can make the perceived image clearer. Using unusual figure/ground relationships can add interest and subtly to an image.

11.3 I know how it should behave

Interactions with everyday things influence a user's cognitive biases when performing actions.

Consideration: User experience designers should think about different aspects of interaction design. The UX designer can create prototypes that illustrate the desired interactions or establish a set of interaction design flows and storyboards.

11.4 I know what it should be named

It is common knowledge that application users tend to scan views for content, rather than reading and evaluating all the content. This scanning pattern makes users locate keywords.

Consideration: Taxonomy and nomenclature are essential for taking advantage of this cognitive bias. Using familiar/recognizable nomenclature is one of the most essential parts of UX design.

APPENDIX VII

APPLE HIG USER EXPERIENCE GUIDELINES

This Appendix presents Apple HIG user experience guidelines based on Apple HIG document (Apple, 2013).

According to Human Interface Guideline, "The user experience of iOS-based devices revolves around streamlined interaction with content that people care about". The following guidelines apply to APPs that run on all iOS-based devices.

1. Focus on the Primary Task

Focus on the primary task, determining the most relevant content for each context or screen.

2. Elevate the Content that People Care About

Elevate the content that users care about, by designing the app's UI as a subtle frame for the information they are interested in; for example:

a. Minimize the number and prominence of controls, in order to decrease their weight in the UI.

b. Subtly customize the controls, so that they integrate with app's graphical style and can be discovered and understood without being conspicuous.

c. Fade controls for a little while after the user has stopped interacting with them, and display them again when the user taps the screen, to help him concentrate more

3. Think Top Down

The top of the screen being the most visible to users and the easiest to reach because they tend to hold the device in their hands; for example:

a. Place the most frequently used (usually higher level) information near the top, and in the following order: from general to specific, and from high level to low level.

4. Give People a Logical Path to Follow

Give users a logical path to follow, so that they know where they are in the app and receive confirmation that they are on the right path.

a. The path should be predictable, and markers, such as the back button, should be provided to inform users where they are and how to retrace their steps.

b. Provide only one path to a screen in most cases; if a screen needs to be accessible in different circumstances, provide a "modal" view as well.

5. Make Usage Easy and Obvious

Strive to make the app instantly understandable to users, so that they do not have to spend time figuring out how it works; for example:

a. Make the primary function of the app immediately apparent, by minimizing the number of controls from which the user must choose, and use standard controls and gestures appropriately and consistently, so that they behave the way the user expects them to.

b. Make the app consistent with the usage paradigms of built-in APPs, with the same screen navigation hierarchy, content listing style, and mode switching capability using the tab bar.

6. Use User-Centric Terminology

User-centered terminology should be used in applications. for example:

a. Use understandable terminology, that is, words and phrases that are appropriate for the targeted user groups, in all text-based communications.

b. Describe dates accurately in the UI, avoiding informal terms such as “today” or “tomorrow”, which may not reflect the user’s current location.

7. Minimize the Effort Required for User Input

Minimize the effort required for user input, as it is time-consuming and requires the user’s attention, whether it is tapped in or keyed in; for example:

a. Balance the user input requested with what the app offers the user in return, providing as much information or functionality as possible for each piece of information entered by the user.

b. Make choices easy for the user, e.g. by providing a table view or a list picker component instead of a text field.

c. Obtain information from the device when it makes sense to do so, so that users are not obliged to provide information that is readily available by the app.

8. Downplay File-Handling Operations

Downplay file handling operations, so that users are minimally aware of the existence of a file system on an iOS device, unless they are creating or manipulating files themselves.

Users should not be prompted to think about file metadata or locations.

a. Ensure that users can manage documents without opening iTunes on their computer, by using iCloud, for example, to help them access their content on all their devices, to the possible extent.

9. Enable Collaboration and Connectedness

Enhance the app by enabling users to collaborate and connect with others in accordance with the mission of iOS devices, which, even though they are personal devices, encourage sharing with others; for example:

a. Ensure that users can easily share information that is important to them, like their location, opinions, and high game scores, when it is appropriate

10. De-emphasize Settings

a. Include settings about preferred app behaviors and information that users rarely want to change in the app only when it is appropriate to do so.

b. Allow users to set their preferred behaviors by using the configuration options in the app.

c. Offer configuration options in the main UI, or on the back of a view in iPhone applications: to decide which location makes sense, and to determine whether or not the options involve a primary task and how often users might want to set them.

11. Brand Appropriately

a. Present the brand colors or images in a subtle and understated way for greatest effect.

b. Avoid taking space away from the content that users care about and using it to display branding assets.

12. Make Search Quick and Rewarding

Make searching quick and rewarding, by making the search function the primary one, and following the guidelines below to ensure that it performs well; for example:

a. Build app data indices, so that the app is always ready to be searched.

b. Live-filter local data, so that the app can display results more quickly, narrowing them as the user continues to type.

c. Filter remote data while the user types when possible, informing him that he can opt out if the response time is likely to delay the results by more than a second or two.

d. Display the search bar above a list, or the index for a list, which is where users are accustomed to finding it in MS Outlook Contacts and other APPs.

e. Feature the search function as a distinct mode if it is a primary function in the app, and only provide a search tab in particular circumstances.

f. Display placeholder content and partial results as they become available to give users prompt access.

g. Provide a scope bar if the data sort naturally into different categories, as this allows users to specify locations or rules in a search, or to filter objects by specific criteria.

13. Entice and Inform with a Well-Written Description

An App Store description is an excellent opportunity to communicate with potential users. In addition to accurately describing the application and highlighting the qualities users might appreciate the most, follow these guidelines:

a. Ensure that all spelling, grammatical, and punctuation errors are corrected, to avoid creating a negative impression of an app's quality.

b. Keep all-capital-letter words to a minimum, as they can make text very difficult to read.

c. Describe specific bug fixes that customers have been waiting for in the description of a new version of an app.

14. Be Succinct

a. Convey information in a condensed, headline-type style so that users can absorb it quickly and easily.

b. Give short labels or well-understood symbols to controls so that users know what they are doing at a glance.

15. Use UI Elements Consistently

Use UI elements consistently, as users expect standard views and controls to look and behave in the same way across APPs; for example:

a. Follow usage recommendations for standard UI elements so that users can depend on previous experience to help them learn to use a new app.

b. Avoid radically changing the appearance of a control that performs a standard action, as users will spend time discovering how to use them and wonder what, if anything, this control does that the standard one does not.

c. Never use the standard buttons and icons to mean something else.

16. Consider Adding Physicality and Realism

Consider adding a physical, realistic dimension to the app when appropriate; for example:

a. Enhance scenes or enlarge objects, if appropriate, to communicate with users and to express the essence of the app, as this can convey more meaning than a faithful likeness.

b. Use appropriate animation to enhance realism in an app, bearing in mind that users will accept artistic license in appearance, but they may feel disoriented when they see movement that appears to defy the laws of physics.

17. Delight People with Stunning Graphics

Delight users with rich, beautiful, and engaging graphics, as they draw users into the app and make the simplest task rewarding, as well as helping to build the app's brand; for example:

a. Replicate the look of high-quality or precious materials, taking the time to make sure the material looks realistic and valuable.

b. Create beautiful, high-resolution artwork and icons from the start, rather than scaling up the quality later.

18. Handle Orientation Changes

Allow orientation changes, as these are often expected by iOS device users:

a. Maintain focuses on the primary content in all orientations, so that users feel they have control over the app and the content they care about.

19. Make Targets Fingertip-Size

Make targets fingertip-size to ensure that users can comfortably use the app, following these guidelines:

a. Tappable elements in an app have a target area of about 44 x 44 points, as this size is necessary for ease of use.

20. Use Subtle Animation to Communicate

Use subtle animation to communicate, because it is highly effective, as long as it does not get in the way of the users' tasks or slow them down; for example:

a. Keep animation consistent with that built into APPs when appropriate, as users are accustomed to the subtle animation they contain.

b. Use animation consistently throughout the application so that users can rely on the experience it gives them.

21. Support Gestures Appropriately

Support gestures adequately and predictably, as iOS device users use gestures to interact with their iOS devices and associate certain behaviors with specific gestures; for example:

a. Avoid changing the actions associated with the standard gestures that users know.

b. Assign complex gestures, or less common ones like swipe or pinch open, as shortcuts to expedite a task, not as the only way to perform a task.

22. Ask People to Save Only When Necessary

Ask users to save only when necessary, as the app, not the user, should save the data, and do so automatically.

23. Make Modal Tasks Occasional and Simple

Make "modal" tasks simple and infrequent, minimizing, when possible, how often the user must be in a modal environment to perform a task or provide a response.

a. Always provide an obvious and safe way to exit a "modal" task, to reassure users that their work is safe when they dismiss a modal view.

24. Start Instantly

Present useful content immediately, as it is often said that users spend no more than a minute or two evaluating a new app.

a. Display a launch image that closely resembles the first screen of the app, to decrease the app’s perceived start time.

b. Avoid displaying an About window or a splash screen, to ensure that users are not prevented from using the app immediately.

c. Delay the login requirement for as long as possible, to enable users to navigate through much of the app and access some of its functionality without logging in.

d. Ensure that, when an app restarts, its state is restored, so that users do not have to remember how they had reached it in the first place.

25. Do not quit programmatically

Try to prevent the app from quitting programmatically, as iOS device users tend to interpret this as crashing.

26. A License Agreement or Disclaimer is displayed when it is necessary

Include an end-user license agreement (EULA) or a disclaimer when necessary, as it will be displayed by the App Store so that users can read it before they acquire the app.

APPENDIX VIII

STANDARDIZED USABILITY QUESTIONNAIRES

This Appendix presents the standardized usability questionnaires that can be used in user-based application usability evaluation according to Sauro and R.Lewis (2012).

1. Post-study questionnaires

Post-study questionnaires applied after application used by the users.

1.1 QUIS (Questionnaire for user interaction satisfaction)

The QUIS assess users' subjective satisfaction with particular aspects of the human–computer interface. QUIS includes "a demographic questionnaire, a measure of overall system satisfaction along six scales, and hierarchically organized measures of nine specific interface factors (screen factors, terminology and system feedback, learning factors, system capabilities, technical manuals, online tutorials, multimedia, teleconferencing, and software installation)".

1.2 SUMI (Software usability measurement inventory)

The SUMI is a standardized questionnaire product of the Human Factors Research Group (HFRG) at University College Cork in Ireland and includes the following criteria:

- This software responds too slowly to inputs.
- I would like to recommend this software to my colleagues.
- The instructions and prompts are helpful.
- The software has at some time stopped unexpectedly.
- Learning to operate this software initially is full of problems.
- I sometimes don't know what to do next with this software.

- I enjoy my sessions with this software.
- I find that the help information given by this software is not very useful.
- If this software stops, it is not easy to restart it.
- It takes too long to learn the software commands.
- I sometimes wonder if I am using the right command.
- Working with this software is satisfying.
- The way that system information is presented is clear and understandable
- I feel safer if I use only a few familiar commands or operations.
- The software documentation is very informative.
- This software seems to disrupt the way I normally like to arrange my work.
- Working with this software is mentally stimulating.
- There is never enough information on the screen when it is needed.
- I feel in command of this software when I am using it.
- I prefer to stick to the facilities that I know best.
- I think this software is inconsistent.
- I would not like to use this software every day.
- I can understand and act on the information provided by this software.
- This software is awkward when I want to do something that is not standard.
- There is too much to read before you can use the software.
- Tasks can be performed in a straightforward manner using this software.

- Using this software is frustrating.
- The software has helped me overcome any problems I have had to use it.
- The speed of this software is fast enough.
- I have to go back to look at the guides.
- It is obvious that user needs have been fully taken into consideration.
- There have been times in using this software when I have felt quite tense.
- The organization of the menu or information lists seems quite logical.
- The software allows the user to be economic of keystrokes.
- Learning how to use new functions is difficult.
- There are too many steps required to get something to work.
- I think this software has made me have a headache on occasion.
- Error prevention messages are not adequate.
- It is easy to make the software do exactly what users want.
- I will never learn to use all that is offered in this software.
- The software has not always done what I was expecting.
- The software has a very attractive presentation.
- Either the amount or quality of the help information varies across the system
- It is relatively easy to move from one part of a task to another.
- It is easy to forget how to do things with this software.
- This software occasionally behaves in a way, which cannot be understood.

- This software is very awkward.
- It is easy to see at a glance what the options are at each stage.
- Getting data files in and out of the system is not easy.
- I have to look for assistance most times when I use this software.

1.3 PSSUQ (Post-study system usability questionnaire)

The PSSUQ is a questionnaire designed to assess users' perceived satisfaction with computer systems or applications.

1.4 SUS (Software usability scale)

The SUS is a questionnaire for end-of-test subjective assessments of usability. The SUS accounted for 43% of post-test questionnaire usage in a recent study of a collection of unpublished usability studies (Sauro and Lewis, 2009).

2. Post-task questionnaires

Post-study questionnaires are essential tools in the usability examiner's toolbox, but they evaluate satisfaction at a relatively high level. This can be a strength when comparing general satisfaction with competitors or different versions of an application, but is a weakness when seeking more detailed diagnoses of problem areas in a UI. To address this weakness, many examiners perform a quick assessment of perceived usability immediately after participants complete each task or scenario in a usability study.

This section describes a variety of commonly used post-task questionnaires.

2.1 ASQ (After-scenario questionnaire)

The ASQ is a three-item questionnaire that uses the same format as the PSSUQ, probing overall ease of task completion, satisfaction with completion time, and satisfaction with support information. The overall ASQ score is the average of the responses to these items.

2.2 SEQ (Single ease question)

The SEQ simply asks users to evaluate the overall ease of completing a task.

2.3 SMEQ (Subjective mental effort question)

The SMEQ is a single-item questionnaire with a rating scale from 0 to 150 with nine verbal labels ranging from “Not at all hard to do” (just above 0) to “Tremendously hard to do” (just above 110).

2.4 ER (Expectation ratings)

The ER addresses the relationship between how easy or challenging a user found a task to be after performing it relative to how they perceived it before beginning the task. The expectation rating procedure uses a variation of the SEQ, getting participants to rate the expected difficulty of all of the tasks planned for a usability study before doing any of the tasks (the expectation ratings), then collecting the post-task rating in the usual way after the completion of each task (the experience rating).

2.5 UME (Usability magnitude estimation)

In a usability evaluation context, the aim of UME is to get a measurement of usability that enables ratio measurement, so a task (or product) with a perceived difficulty of 100 is perceived as twice as difficult as a task (or product) with a perceived difficulty of 50.

3. Questionnaires for assessing perceived usability of websites

After the Web had started to get popular as a tool to share information and to conduct commerce, questionnaires designed more specifically for the assessment of the perceived usability of websites appeared in the literature. Following sub-sections describe the website usability related questionnaires.

3.1 WAMMI (Website analysis and measurement inventory)

The WAMMI has a set of 20 five-point items, still covering five sub-scales (Attractiveness, Controllability, Efficiency, Helpfulness, and Learnability) and a global measure.

3.2 SUPR-Q (Standardized universal percentile rank questionnaire)

The SUPR-Q is a rating scale designed to measure perceptions of usability, credibility/ trust, appearance, and loyalty for websites.

4. Other questionnaires of interest

Recently, there have been a number of other publications of questionnaires designed for the assessment of usability. The focus of these researches has ranged from assessment of perceived quality and satisfaction to perceived usability. The following subsections describe these questionnaires.

4.1 CSUQ (Computer system usability questionnaire)

The CSUQ is a variant of the PSSUQ, developed to permit the collection of a vast number of completed questionnaires and to see if the factor structure found for the PSSUQ in a usability testing setting would stay the same in a mailed survey. The emergence of the same factors would demonstrate the potential usefulness of the questionnaire across different user groups and research settings.

4.2 USE (Usefulness, satisfaction, and ease of use)

The USE is a 30-item questionnaire designed to capture information about Usefulness, Ease of Use, Ease of Learning, and Satisfaction.

4.3 UMUX (Usability metric for user experience)

The UMUX aims to get a measurement of perceived usability consistent with the SUS but using fewer items that more closely conformed to the ISO definition of usability (effective, efficient, satisfying).

4.4 HQ (Hedonic quality)

The HQ has seven seven-point bipolar items. Originally in German, then translated into English, the HQ bipolar scale anchors are:

- HQ1: interesting—boring
- HQ2: costly—cheap
- HQ3: exciting—dull
- HQ4: exclusive—standard
- HQ5: impressive—nondescript
- HQ6: original—ordinary
- HQ7: innovative—conservative

4.5 ACSI (American customer satisfaction index)

The ACSI model includes perceived quality, perceived value, and customer expectations driving customer satisfaction, which in turn affects customer loyalty and complaints.

4.6 NPS (Net promoter score)

The NPS uses a single likelihood to support question ("How likely is it that you would recommend our company to a friend or colleague?") with 11 scale steps from zero (not at all likely) to 10 (extremely likely).

4.7 CxPi (Forrester customer experience index)

The CxPi uses responses to three questions designed to address perceived usefulness, usability, and enjoyability. For each question, respondents make choices on a five-point scale (1 = very negative experience to 5 = very positive experience).

4.8 TAM (Technology acceptance model)

According to the TAM, the primary factors that affect a user's intention to use a technology are its perceived usefulness and perceived ease of use. Actual use of technologies is influenced by the intention to use, which is itself affected by the perceived usefulness and usability of the technology. In the TAM, perceived usefulness is the extent to which a person believes a technology will enhance job performance, and perceived ease of use is the degree to which a user believes that using the technology will be effortless.

4.9 MPUQ (Mobile Phone Usability Questionnaire)

Ryu (2005) has developed a usability questionnaire for mobile phones that have 124 questions to ask to the user. This questionnaire is designed for mobile phones so considers the challenges in mobile usability.

APPENDIX IX

COMMON USABILITY STUDY SCENARIOS

This Appendix presents the common usability study scenarios that are used during usability evaluation according to Tullis and Albert (2013).

1. Completing a transaction

Many usability studies are aimed at making transactions run as smoothly as possible. These might take the form of a user completing a purchase, registering a new piece of software, or selling stock. A transaction usually has a well-defined beginning and end. In completing a transaction study perhaps the first measure that we will want to examine is task success. Each task is scored as a success or failure. Apparently the tasks need to have a clear end-state, such as reaching a confirmation that the transaction was successful. Other measures that can be examined are as follows: Efficiency, Issues-based measures, and self-reported measures.

2. Comparing applications

It is always useful to know how developed application compares to the competition or previous releases. By making comparisons, we can determine application's strengths and weaknesses and whether improvements have been made from one version to another. The best way to compare different applications or releases is through the use of various measures. The type of measures we choose should be based on the application itself. Some applications aim to maximize efficiency, whereas others try to create an exceptional user experience. Three types of measures are prescribed during application comparison as follows: Task success, Efficiency, and self-reported measures.

3. Evaluating frequent use of the same application

Many applications are intended to be used on a frequent or semi-frequent basis. These applications need to be both easy to use and highly efficient. Hence, measures recommended

for evaluating frequent use of the same application are as follows: Task time, Task success, Efficiency, Learnability and self-reported measures.

4. Evaluating navigation and/or information architecture

Many usability studies focus on improving the navigation and/or information architecture. It may involve making sure that users can instantly and easily find what they are looking for, easily navigate around the application, know where they are within the overall structure, and know what options are available to them. Typically, these studies involve the use of wire-frames or partially functional prototypes because the navigation and information mechanisms and information architecture are so fundamental to the design that they have to be figured out before almost anything else. Measures recommended for evaluating navigation and/or information architecture are as follows: Task success, Errors, and Efficiency.

5. Increasing awareness

Not every design that goes through a usability evaluation is about making something easier or more efficient to use. Some design changes are directed to increasing awareness of a particular piece of content or functionality. This is true for applications that have important but under-utilized functionality. There can be many reasons why something is not noticed or used, including some aspect of the visual design, labeling, or placement. Recommended measures for increasing awareness are: Self-reported measures and Behavioral and physiological measures.

6. Problem discovery

The goal of problem discovery is to identify major usability issues. In some situations, one may not have any preconceived ideas about what the significant usability issues are with an application, but want to know what annoys users. Aforementioned is usually done for an application that is already built but has not gone through usability evaluation before. A problem discovery study also works well as a periodic checkup to get back in touch with how users are

interacting with the application. Recommended measures for problem identification are: Issue-based measures and self-reported measures.

7. Maximizing usability for a critical application

Although some applications strive to be easy to use and efficient, other applications have to be easy to use and efficient. What differentiates a critical application from a noncritical application is that the entire reason for the critical application's existence is for the user to complete a significant task. Not completing that task will have a significant adverse outcome. Measuring the usability of any critical product is essential and following measures should be held: Task success, Errors, and Efficiency.

8. Creating an overall positive user experience

Some applications strive to create an exceptional user experience. It is simply not enough to be usable. These applications need to be engaging, thought-provoking, entertaining, and maybe even slightly addictive. Their popularity usually grows at phenomenal rates. Even though the characteristics of what constitutes a great user experience are subjective, they are still measurable.

Although some performance measures may be useful, what matters is what the user thinks, feels, and says with respect to his or her experience. Recommended measures for creating an overall positive user experience are self-reported measures and Behavioral and physiological measures.

9. Comparing alternative designs

One of the most common types of usability studies involves comparing more than one design alternative. Typically, these types of studies take place early in the design process before the design has been completely developed. Different design teams put together semi-functional prototypes, and one evaluate each design using a predefined set of measures. Asking the same

participant to perform the same task with all designs usually does not yield valuable information, even when counterbalancing design and task order. Recommended measures for comparing alternative designs are Task success, Task Time, Issues-based measures and Self-reported measures.

APPENDIX X

LIST OF CITED-BY PUBLICATIONS

This chapter lists the publications that cited author's publications.

1. The State of the Art of Mobile Usability Evaluation

[1] Lew, Philip, and Luis Olsina. "Relating User Experience with MobileApp Quality Evaluation and Design." In Current Trends in Web Engineering, pp. 253-268. Springer International Publishing, 2013.

[2] AlRoobaea, Roobaea S., Ali H. Al-Badi, and Pam J. Mayhew. "Generating a Domain Specific Inspection Evaluation Method through an Adaptive Framework: A Comparative Study on Educational Websites." International Journal of Human Computer Interaction (IJHCI) 4, no. 2, 2013.

[3] AlRoobaea, Roobaea S., Ali H. Al-Badi, and Pam J. Mayhew. "A framework for generating a domain specific inspection evaluation method: A comparative study on social networking websites." In Science and Information Conference (SAI), 2013, pp. 757-767. IEEE, 2013.

[4] Gündüz, Feyza, and Al-Sakib Khan Pathan. "On the Key Factors of Usability in Small-sized Mobile Touch-Screen Application." Int. J. Multimed. Ubiquitous Eng 8, no. 3 (2013): 115-138.

[5] Nayebi, F.; Desharnais, J.-M.; Abran, A., "An Expert-Based Framework for Evaluating iOS Application Usability, *Joint Conference of the 23rd International Workshop on Software Measurement and the Eighth International Conference on Software Process and Product Measurement (IWSM-MENSURA)*, pp.147,155, 23-26 Oct. 2013 Ankara doi: 10.1109/IWSM-Mensura.2013.30.

[6] AlRoobaea, Roobaea S., Ali H. Al-Badi, and Pam J. Mayhew. "Generating an Educational Domain Checklist through an Adaptive Framework for Evaluating Educational Sys-

tems." IJACSA) International Journal of Advanced Computer Science and Applications 4, no. 8 (2013).

[7] Alroobaea, Roobaea S., Ali H. Al-Badi, and Pam J. Mayhew. "Generating a Domain Specific Checklist through an Adaptive Framework for Evaluating Social Networking Websites." In IJACSA) International Journal of Advanced Computer Science and Applications, Special Issue on Extended Papers from Science and Information Conference, p. 25. 2013.

[8] Dundar, Bahadir, Nejat Yumusak, and Samet Arsoy. "GuidedBased Usability Evaluation on Mobile Websites." ICIW 2013, The Eighth International Conference on Internet and Web Applications and Services, pp. 212-217. June 23-28 2013, Rome.

[9] de Morais Barroca Filho, Itamir, and Gibeon Soares de Aquino Junior. "MetamorphosIS: A Process for Development of Mobile Applications from Existing Web-Based Enterprise Systems." In Computational Science and Its Applications–ICCSA 2014, pp. 17-30. Springer International Publishing, 2014.

[10] Saeed, Fatima, Dost Muhammad Khan2 Najia Saher, Faisal Shahzad, and Nouman Ammer. "A NOVEL FRAMEWORK FOR INTERACTIVE MOBILE APPLICATIONS." Sci.Int. (Lahore), 26(5), 2089-2095, 2014 ISSN 1013-5316; CODEN: SINTE 8 300.

[11] Olsina, Luis, Lucas Santos, and Philip Lew. "Evaluating Mobileapp Usability: A Holistic Quality Approach." Lecture Notes in Computer Science Volume 8541, 2014, pp 111-129. Springer International Publishing, 2014.

[12] de Lima Salgado, André, and André Pimenta Freire. "Heuristic Evaluation of Mobile Usability: A Mapping Study." In Human-Computer Interaction. Applications and Services, pp. 178-188. Springer International Publishing, 2014.

[13] Alqahtani, Mohammed A., Obead Alhadreti, Roobaea S. AlRoobaea, and Pam J. Mayhew. "Investigation into the Impact of the Usability Factor on the Acceptance of Mobile Transac-

tions: Empirical Study in Saudi Arabia." International Journal of Human Computer Interaction (IJHCI) 6, no. 1 (2015).

[14] Martina, Andrea. "Virtual Heritage: new technologies for edutainment." PhD diss., Politecnico di Torino, Italy, 2014.

[15] Júnior, Gibeon Soares de Aquino, and Itamir de Morais Barroca Filho. "SIGAA Mobile A successful experience of constructing a mobile application from a existing web system." SEKE 2013 June 27-29, 2013, pp. 510-515.

[16] Petea, Balázs, and Rob Brennanb. "A User Study of Visual Linked Data Query and Exploration in Mobile Devices."

[17] Alshehri, Fayez, Mark Freeman, and Alison E. Freeman (2013). "Are you smart enough for your smart phone? A cognitive load comparison." 24th Australasian Conference on Information Systems (pp. 1-11). Australia: RMIT University.

[18] Yang, Wei, and Xiao Yu. "AT-EASE: A Tool for Early and Quick Usability Evaluation of Smartphone Application."

[19] Umar, Muhammad Aminu, and Masitah Ghazali. "Investigation into Usability Attributes for Embedded Systems Testing." International Journal of Software Engineering and Technology 1, no. 2 (2014).

[20] Vasconcelos, Pattrisce. "Fatores-chave de sucesso na adoção de aplicativos móveis de táxi."

[21] de Morais Barroca Filho, Itamir, and Gibeon Soares de Aquino Junior. "A metamorfose dos sistemas de informação na era da computação móvel." Revista Brasileira de Administração Científica 4, no. 2 (2013): 6-17.

[22] Cortés, Hurtado, and Luini Leonardo. "Desarrollo de una metodología de evaluación de usabilidad de interfaces humano-máquina (IHM) para la mejora del Proceso de toma de

decisiones en tareas de supervisión industrial." PhD diss., Universidad Nacional de Colombia-Sede Manizales, Columbia.

[23] Vasquez, Gustavo A., "Évaluation de l'utilisabilité des applications ios au niveau design et de l'interface humain (human interface) : une étude statistique portant sur 40 applications pour le iphone.", Rapport de projet, École de technologie supérieure, maîtrise en génie, concentration ti, 2013.

[24] Ouhbi, Sofia, José Luis Fernández-Alemán, José Rivera Pozo, Manal El Bajta, Ambrosio Toval, and Ali Idri. "Compliance of Blood Donation Apps with Mobile OS Usability Guidelines." Journal of medical systems 39, no. 6 (2015): 1-21.

[25] Pitassi, Emanuela. "Social and Semantic Contexts in Tourist Mobile Applications." (2015).

[26] Barroca Filho, Itamir de Morais, and Gibeon Soares Aquino Júnior. "Development of mobile applications from existing web-based enterprise systems." International Journal of Web Information Systems 11, no. 2 (2015).

[27] Silva, Williamson, Natasha M. Costa Valentim, and Tayana Conte. "Integrating the Usability into the Software Development Process." ICEIS 2015 - 17th International Conference on Enterprise Information Systems.

[28] Hussain, Azham Bin, Sharaf Aldeen Abdulkadhum Abbas, Mustafa Sabah Abdulwaheed, Rammah Ghanim Mohammed, and Adil abdullah Abdulhussein. "Usability Evaluation of Mobile Game Applications: A Systematic Review." International Journal of Computer and Information Technology (ISSN: 2279 – 0764) Volume 04 – Issue 03, May 2015.

[29] Fitrawan, Alif Akbar, Christian Sri kusuma Aditya, and Umi Laili Yuhana. "Pengukuran Kualitas Perangkat Lunak berdasarkan ISO/IEC 25000: Systematic Mapping." Jurnal Manajemen Informatika 4, no. 01 (2015).

[30] Radisavljević, Nikola, and Džana Kujan. "Migrating process automation applications to mobile." Master thesis in Software Engineering, School of Innovation, Design and Engineering Malardalen University, 2015.

[31] Reavis, David. "A Comparison Of Functionality Between Mobile Apps And Browser-Based Applications." Review of Business Information Systems (RBIS) 19, no. 1 (2015): 15-18.

2. An Expert-based Framework for Evaluating iOS Application Usability

[1] Coutu-Nadeau, Charles. "Evaluating the usability of diabetes management iPad applications." PhD diss., WEILL MEDICAL COLLEGE OF CORNELL UNIVERSITY, 2014.

[2] Alqahtani, Mohammed A., Obead Alhadreti, Roobaea S. AlRoobaea, and Pam J. Mayhew. "Investigation into the Impact of the Usability Factor on the Acceptance of Mobile Transactions: Empirical Study in Saudi Arabia." International Journal of Human Computer Interaction (IJHCI) 6, no. 1 (2015).

[3] Vasquez, Gustavo A., "Évaluation de l'utilisabilité des applications ios au niveau design et de l'interface humain (human interface) : une étude statistique portant sur 40 applications pour le iphone.", Rapport de projet, École de technologie supérieure, maîtrise en génie, concentration ti, 2013.

BIBLIOGRAPHY

Abran, A. *Software Metrics and Software Metrology*. Wiley-IEEE Computer Society Press, 2010. ISBN 0470597208, 9780470597200.

Abran, A., Khelifi, A., Suryn, W., and Seffah, A. Consolidating the iso usability models. *Proceedings of 11th International Software Quality Management Conference and the 8th Annual INSPIRE Conference*, p. 23-25, 2003.

Akaike, H. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, p. 716-723, December 1974. ISSN 0018-9286.

Al-Qaimari, G. and Fernando, S. Location-aware applications: Evaluating the ease of use and ease of learning. *Proceedings of the 2006 International Conference on Wireless Communications and Mobile Computing*, IWCMC '06, p. 1283–1288, New York, NY, USA, 2006. ACM. ISBN 1-59593-306-9. doi: 10.1145/1143549.1143807. <http://doi.acm.org/10.1145/1143549.1143807>.

Alexander, T., Sclick, C., Sievert, A., and Leyk, D. Assessing human mobile computing performance by fitts' law. *Handbook of Research on User Interface Design and Evaluation for Mobile Technology*, p. 830-846, July 2008. doi: 10.4018/978-1-59904-871-0.ch049.

Alpaydin, E. *Introduction to Machine Learning*. The MIT Press, ed. 3rd, 2014.

Anuar, N. B., Kuen, L. N., Zakaria, O., Gani, A., and Wahab, A. W. A. Usability and performance of secure mobile messaging: M-pki. *WSEAS Transactions on Information Science and Applications*, p. 179-189, 2009.

Apple. *iOS Human Interface Guidelines*. Apple Inc., 2013.

Apple. Apple app store, 2015. <https://www.apple.com/ca/itunes/charts/free-apps/>.

Azevedo-Filho, A. and Shachter, R. D. Laplace's method approximations for probabilistic inferencein belief networks with continuous variables. *Proceedings of the Tenth International Conference on Uncertainty in Artificial Intelligence*, UAI'94, p. 28–36, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc. ISBN 1-55860-332-8. <http://dl.acm.org/citation.cfm?id=2074394.2074399>.

Bahn, S., Lee, C., Jo, J., Suh, W., Song, J., and Yun, M. Incorporating user acceptance into usability evaluation scheme for the user interface of mobile services. *IEEE International Conference on Industrial Engineering and Engineering Management, 2007*, p. 492-496, Dec 2007. doi: 10.1109/IEEM.2007.4419238.

Baker, A. Simplicity. *Stanford Encyclopedia of Philosophy*, (ISSN 1095-5054), July 2012.

Balagtas-Fernandez, F. and Hussmann, H. A methodology and framework to simplify usability analysis of mobile applications. *24th IEEE/ACM International Conference on Automated Software Engineering, 2009. ASE'09*, p. 520-524, 2009.

Bernhaupt, R., Mihalic, K., and Obrist, M. Usability evaluation methods for mobile applications. Lumsden, J., editor, *Handbook of research on user interface design and evaluation for mobile technology*, p. 745–758. IGI Global, Hershey, PA, 2008.

Biel, B., Grill, T., and Gruhn, V. Exploring the benefits of the combination of a software architecture analysis and a usability evaluation of a mobile application. *Journal of Systems and Software*, volume 83, p. 2031–2044, New York, NY, USA, November 2010. Elsevier Science Inc. doi: 10.1016/j.jss.2010.03.079. <http://dx.doi.org/10.1016/j.jss.2010.03.079>.

Billi, M., Burzagli, L., Catarci, T., Santucci, G., Bertini, E., Gabbanini, F., and Palchetti, E. A unified methodology for the evaluation of accessibility and usability of mobile applications. *Universal Access in the Information Society*, p. 337–356, March 2010.

Bishop, C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 0387310738.

Bouckaert, R. R. Bayesian network classifiers in weka for version 3-5-7. *Artificial Intelligence Tools*, p. 369–387, 2008.

Budgen, D., Turner, M., Brereton, P., and Kitchenham, B. Using mapping studies in software engineering. *Proceedings of PPIG 2008, Lancaster University*, p. 195-204, 2008.

Cheikhi, L. and Abran, A. Investigation of the relationships between the software quality models of iso 9126 standard: An empirical study using the taguchi method. *Software Quality Professional, American Society for Quality*, 14(2), March 2012.

Cheikhi, L., Abran, A., and Buglione, L. The isbsg software projects repository: An analysis for the iso 9126 perspective. *Software Quality Management Journal, American Society for Quality*, p. 4-16, 2007.

Cheikhi, L., Abran, A., and Suryn, W. Harmonization of usability measurement in iso software engineering standards. *IEEE International Symposium on Industrial Electronics ISIE'06*, p. 3246-3251, July 2006.

Chua, A. Y. K., Goh, D. H., Lee, C. S., and Tan, K.-T. Mobile alternate reality gaming engine: A usability evaluation. *Seventh International Conference on Information Technology*, January 2010.

Collobert, R. and Bengio, S. Links between perceptrons, mlps and svms. *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, p. 23, New York, NY, USA, 2004. ACM. ISBN 1-58113-838-5. doi: 10.1145/1015330.1015415. <http://doi.acm.org/10.1145/1015330.1015415>.

Concejero, P., Patrocinio, J. C. L., and Merino, D. T. Usability evaluation of mobile services. *12th International Conference on Intelligent Service Delivery Networks. ICIN 2008*, p. 6, Cirencester, UK, 2008. Telefonica Investig. y Desarrollo, Madrid, Spain.

Coronel, N. O. M., Hernandez, J. O., Sanchez, S. S., and Serna, J. G. G. Integration of usability into the development of mobile computing environments applied to contextual location based services (lbs). *Electronics, Robotics and Automotive Mechanics Conference*, January 2011.

Coursaris, C. K. A meta-analytical review of empirical mobile usability studies. *Journal of Usability Studies*, p. 117–171, 2011.

Coursaris, C. K. and Kim, D. J. A qualitative review of empirical mobile usability studies. *Proceedings of the Twelfth Americas Conference on Information Systems*, p. 1–14, 2006.

Cox, R. T. Probability, frequency and reasonable expectation. *American Journal of Physics*, p. 1-13, 1946. doi: http://dx.doi.org/10.1119/1.1990764. <http://scitation.aip.org/content/aapt/journal/ajp/14/1/10.1119/1.1990764;jsessionid=2i62l09g1gjjs.x-aip-live-02>.

Crease, M. and Longworth, R. Mobile evaluations in a lab environment. *Mobile Computing: Concepts, Methodologies, Tools, and Applications. IGI Global*, p. 2042–2060, July 2009.

Crossan, A., Murray-Smith, R., Brewster, S., and Musizza, B. Instrumented usability analysis for mobile devices. *International Journal of Mobile Human Computer Interaction (IJMHCI), 1(1), 1-19*, July 2009.

Das, B. and Mohanty, S. Service usability and users' satisfaction in india: An exploratory study on mobile phone users. *ICFAI Journal of Services Marketing, ICFAI University Press*, p. 1–15, December 2007.

Dix, A., Finlay, J., Abowd, G., and Beale, R. *Human-Computer Interaction*. Prentice-Hall, 1993.

Dix, A., Bertini, E., Catarci, T., Gabrielli, S., Kimani, S., and Santucci, G. Appropriating heuristic evaluation methods for mobile computing. *Handbook of Research on User Interface Design and Evaluation for Mobile Technology*, February 2007.

Duh, H. B.-L. and Tan, G. C. B. Usability evaluation for mobile device: A comparison of laboratory and field tests. *8th International Conference on Human Computer Interaction with Mobile Devices and Services*, p. 12-15, 2006.

Fenton, N. E. and Pfleeger, S. L. *Software Metrics: A Rigorous and Practical Approach*. PWS Publishing Co., Boston, MA, USA, ed. 2nd, 1998. ISBN 0534954251.

Fernández-Delgado, M., Cernadas, Senén Barro, E., and Amorim, D. Do we need hundreds of classifiers to solve real world classification problems? *Journal of Machine Learning Research*, p. 3133-3181, 2014. <http://jmlr.org/papers/v15/delgado14a.html>.

Fetaji, M., Dika, Z., and Fetaji, B. Usability testing and evaluation of a mobile software solution: A case study. *Proceedings of the ITI 2008 30th Int. Conf. on Information Technology Interfaces*, June 23-26 2008.

Fibonnaci. Fibonnaci sequence. <http://mathworld.wolfram.com/FibonacciNumber.html>.

Fitts, P. M. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, p. pp. 381–391, June 1954.

Fleiss, J. L., Levin, B., and Paik, M. C. *Statistical Methods for Rates and Proportions*. Wiley, ed. Third, 2004.

Gafni, R. Usability issues in mobile-wireless information systems. *Issues in Informing Science and Information Technology*, 6, 2009.

Gerhardt-Powals, J. Cognitive engineering principles for enhancing human-computer performance. *International Journal of Human-Computer Interaction, 8(2), 189-21*, 1996.

Giffin, A. and Caticha, A. Updating probabilities with data and moments. *Proceedings of 27th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering*, p. 74–84, 8-13 July 2007. doi: 10.1063/1.2821302.

Google. *Android User Interface Guidelines*. Google Inc., 2014.

Gutlein, M., Frank, E., Hall, M. A., and Karwath, A. Large-scale attribute selection using wrappers. *IEEE Symposium on Computational Intelligence and Data Mining, CIDM '09*, p. 332–339, 2009.

Ham, D.-H., Heo, J., Fossick, P., Wong, W., Park, S., Song, C., and Bradley, M. Framework and model of usability factors of mobile phones. *Handbook of Research on User Interface Design and Evaluation for Mobile Technology. IGI Global*, p. 877–896, July 2008.

Han, H. and Jeong, W. Usability study on mobile web newspaper sites. *ASIST 2011, October 9-13, 2011, New Orleans, LA, USA*, January 2011.

Han, J., Kamber, M., and Pei, J. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, ed. 3rd, 2011. ISBN 0123814790, 9780123814791.

Hashim, A. S., Ahmad, W. F. W., and Ahmad, R. Mobile learning course content application as a revision tool: The effectiveness and usability. *International Conference on Pattern Analysis and Intelligent Robotics 28-29 June 2011, Putrajaya, Malaysia*, June 2011.

Hegarty, R. and Wusteman, J. Evaluating ebscohost mobile. *Library Hi Tech*, p. 320–333, 2011.

Heo, J., Ham, D.-H., Park, S., Song, C., and Yoon, W. C. A framework for evaluating the usability of mobile phones based on multi-level, hierarchical model of usability factors. *Interacting with Computers*, p. 263–275, 2009.

Hick, W. E. On the rate of gain of information. *Quarterly Journal of Experimental Psychology*, (4:11-26), 1952.

Hoegh, R. T., Kjeldskov, J., Skov, M. B., and Stage, J. A field laboratory for evaluating in situ. *Handbook of Research on User Interface Design and Evaluation for Mobile Technology (pp. 982-996)*, July 2008.

Hummel, K. A., Hess, A., and Grill, T. Environmental context sensing for usability evaluation in mobile hci by means of small wireless sensor networks. *Proceedings of the 6th International Conference on Advances in Mobile Computing and Multimedia*, MoMM '08, p. 302-306, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-269-6. doi: 10.1145/1497185.1497248. <http://doi.acm.org/10.1145/1497185.1497248>.

Hussain, A. and Kutar, M. Usability metric framework for mobile phone application. *The 10th Annual Conference on the Convergence of Telecommunications, Networking and Broadcasting, 22-23 June*, July 2009.

Hyman, R. Stimulus information as a determinant of reaction time. *Journal of Experimental Psychology*, (45:188-196), 1953.

ISO/IEC12207. Systems and software engineering — software life cycle processes. *International Organization for Standardization (ISO), Geneva*, 2008.

ISO/IEC13407. Human-centered design processes for interactive systems. *International Organization for Standardization (ISO), Geneva*, 1999.

ISO/IEC15939. Systems and software engineering – measurement process. *International Organization for Standardization (ISO), Geneva*, 2007.

ISO/IEC25010. Systems and software engineering - systems and software quality requirements and evaluation (square) - system and software quality models. *International Organization for Standardization (ISO), Geneva*, 2011.

ISO/IEC9126. Software product evaluation - quality characteristics and guidelines for the user. *International Organization for Standardization (ISO), Geneva*, 2001.

ISO/IEC9241. Ergonomics requirements for office with visual display terminals (vdts). *International Organization for Standardization (ISO), Geneva*, 1997.

Jambon, F., Golanski, C., and Pommier, P.-J. Meta-evaluation of a context-aware mobile device usability. *International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, 2007. UBICOMM'07*, p. 21–26, 2007.

Ji, Y. G., Park, J. H., Lee, C., and Yun, M. H. A usability checklist for the usability evaluation of mobile phone user interface. *International Journal of Human-Computer Interaction*, p. 207–231, July 2006.

Juhasz, Z., Arato, A., Bognar, G., Buday, L., Eberhardt, G., Markus, N., Mogor, E., Nagy, Z., and Vaspori, T. Usability evaluation of the most mobile assistant (slattalker). *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence*

and Lecture Notes in Bioinformatics), p. 1055–1062, Linz, Austria, 2006. Department of Information Systems, Pannon University, Veszprem, Hungary, Springer Verlag.

Kitchenham, B. Guidelines for performing systematic literature reviews in software engineering, version 2.3. *EBSE Technical Report, Keele University, UK*, 2007.

Klajajevic, V. Cognitive models as usability testing tools. *Handbook of Research on User Interface Design and Evaluation for Mobile Technology*, p. 814–829, July 2008.

Koch, R. *The 80/20 Principle: The Secret of Achieving More with Less*. Nicholas Brealey Publishing, 2001.

Kunjachan, M. A. C. Evaluation of usability on mobile user interface. *Master Thesis, University Of Washington, Bothell*, May 2010.

Lee, Y. S., Basapur, S., Zhang, H., Guerrero, C., and Massey, N. Usability evaluation of beep-to-the-box. *Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '10, p. 345–348, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-835-3. doi: 10.1145/1851600.1851662. <http://doi.acm.org/10.1145/1851600.1851662>.

Liu, B. Sentiment analysis and subjectivity. *Handbook of Natural Language Processing, Second Edition. Taylor and Francis Group, Boca*, 2010.

Livio, M. *The Golden Ratio: The Story of Phi, The World's Most Astonishing Number*. Number ISBN 0-7679-0815-5. New York: Broadway Books, 2003.

Maly, I., Mikovec, Z., and Vystrcil, J. Interactive analytical tool for usability analysis of mobile indoor navigation application. *3rd International Conference on Human System Interactions (HSI)*, p. 259–266, Rzeszow, Poland, 2010. Czech Technical University in Prague, Faculty of Electrical Engineering, Prague, Czech Republic, IEEE.

Meech, S. *Rule of thirds*. Number ISBN 0-7134-8987-1. Sterling Publishing, 2007.

Miranda, E. The use of reliability growth models in project management. *EEE System and Software Reliability Engineering Conference, Paderborn, Germany*, 1998.

Moritz, F. and Meinel, C. Mobile web usability evaluation-combining the modified think aloud method with the testing of emotional, cognitive and conative aspects of the usage of a web application. *Proceedings - 9th IEEE/ACIS International Conference on Computer and Information Science, ICIS 2010*, p. 367–372, 2010.

Nayebi, F., Desharnais, J., and Abran, A. The state of the art of mobile application usability evaluation. *25th IEEE Canadian Conference on Electrical and Computer Engineering, CCECE 2012, Montreal, QC, Canada, April 29 - May 2, 2012*, p. 1–4, 2012. doi: 10.1109/CCECE.2012.6334930. <http://dx.doi.org/10.1109/CCECE.2012.6334930>.

Nayebi, F., Desharnais, J., and Abran, A. An expert-based framework for evaluating ios application usability. *2013 Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement, Ankara, Turkey, October 23-26, 2013*, p. 147–155, 2013. doi: 10.1109/IWSM-Mensura.2013.30. <http://dx.doi.org/10.1109/IWSM-Mensura.2013.30>.

Nielsen, J. *Usability Engineering*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1994. ISBN 0125184050.

Passani, L. Building usable wireless applications for mobile phones. *Mobile HCI, LNCS 2411, Springer-Verlag*, p. 9–20, January 2002.

Petersen, K., Feldt, R., Mujtaba, S., and Mattsson, M. Systematic mapping studies in software engineering. *12th International Conference on Evaluation and Assessment in Software Engineering (EASE), Department of Informatics, University of Bari, Italy*, June 2008.

Pham, T. P., Razikin, K., Goh, D. H.-L., Kim, T. N. Q., Quach, H. N. H., Theng, Y.-L., Chua, A. Y. K., and Lim, E.-P. Investigating the usability of a mobile location-based annotation system. *Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia*, p. 313. ACM Press, 2010.

Platt, J. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical Report MSR-TR-98-14, Microsoft Research, April 1998. <http://research.microsoft.com/apps/pubs/default.aspx?id=69644>.

Pousttchi, K. and Thurnher, B. Understanding effects and determinants of mobile support tools: A usability-centered field study on it service technicians. *Proceedings of the International Conference on Mobile Business (ICMB'06)*, January 2006.

Qiu, Y. F., Chui, Y. P., and Helander, M. G. Usability analysis of mobile phone camera software systems. *International Conference on Computational Intelligence and Security*, July 2006.

Read, J. C. Using wizard of oz to evaluate mobile applications. *Handbook of Research on User Interface Design and Evaluation for Mobile Technology*, p. 802–813, July 2008.

Ryu, Y. S. Development of usability questionnaires for electronic mobile products and decision making methods, 2005.

Ryu, Y. S. and Smith-Jackson, T. L. Reliability and validity of the mobile phone usability questionnaire (mpuq). *Journal of Usability Studies vol. 2. Issue 1. November*, p. 39–53, November 2006.

Sauro, J. and R.Lewis, J. *Quantifying the User Experience: Practical Statistics for User Research*. Morgan Kaufmann, 2012.

Scholar, G. Citing articles: An expert-based framework for evaluating ios application usability, March 2015a. <https://scholar.google.ca/scholar?oi=bibs&hl=en&cites=17810305529977288272&as_sdt=5>.

Scholar, G. Citing articles: The state of the art of mobile application usability evaluation, March 2015b. <https://scholar.google.ca/scholar?oi=bibs&hl=en&cites=8429109550953644147&as_sdt=5>.

Schultz, D. 10 usability tips & tricks for testing mobile applications. *interactions*, p. 14, November 2006.

Shneiderman, B., Plaisant, C., Cohen, M., and Jacobs, S. *Designing the User Interface: Strategies for Effective Human-Computer Interaction, 5th edition*. Addison-Wesley, 2010.

Statistia. Apple app store statistics, 2015. <http://www.statista.com/statistics/263794/number-of-downloads-from-the-apple-app-store/>.

Stoica, A., Flotakis, G., Raptis, D., Papadimitriou, I., Komis, V., and Avouris, N. Field evaluation of collaborative mobile applications. *Mobile Computing: Concepts, Methodologies, Tools, and Applications (pp. 3251-3269). Hershey, PA: Information Science Reference.*, July 2009.

Streefkerk, J. W., van Esch-Bussemakers, M. P., Neerincx, M. A., and Looije, R. *Evaluating Context-Aware Mobile Interfaces for Professionals*. Handbook of Research on User Interface Design and Evaluation for Mobile Technology (pp. 759-779). Hershey, PA: Information Science, 2008.

Tesoriero, R., Lozano, M., Gallud, J., and Penichet, V. Evaluating the users' experience of a pda-based software applied in art museums. *Proceedings of 3rd International Conference on Web Information Systems and Technologies (Webist 2007)*, p. 351–358, Barcelona, Spain, 2007. Laboratory of User Interaction and Software Engineering, Department of Computer Systems, University ofCastilla-La Mancha, Albacete, Spain.

Tullis, T. and Albert, W. *Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics*. Morgan Kaufmann, ed. Second, 2013.

Turnell, M. d. F. Q. V., Queiroz, J. E. R. d., and Ferreira, D. d. S. *Multilayered Approach to Evaluate Mobile User Interfaces*. Concepts, Methodologies, Tools, and Applications. Mobile Computing: Concepts, Methodologies, Tools, and Applications. IGI Global, July 2009.

Vuolle, M., Kallio, T., Kulju, M., Tiainen, M., Vainio, T., and Wigelius, H. Developing a questionnaire for measuring mobile business service experience. *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, p. 53–62, 2008.

Weinschenk, S. and Barker, D. T. *Designing effective speech interfaces*. Number ISBN 9780471375456. Wiley computer publishing, 2000.

Wikipedia. Cognitive bias, a. <en.wikipedia.org/wiki/Cognitive_bias>.

Wikipedia. Principles based on attention in human computer interaction, b. <http://en.wikipedia.org/wiki/Human\T1\textendashcomputer_interaction>.

Witten, I. H., Frank, E., and Hall, M. A. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, ed. 3rd, 2011. ISBN 0123748569, 9780123748560.

Wright, T., Yoong, P., Noble, J., Cliffe, R., Hoda, R., Gordon, D., and Andreae, C. Usability methods and mobile devices: An evaluation of mofax. *Proceedings of the 4th International Conference on Mobile and Ubiquitous Multimedia*, MUM '05, p. 26–33, New York, NY, USA, 2005. ACM. ISBN 0-473-10658-2. doi: 10.1145/1149488.1149493. <http://doi.acm.org/10.1145/1149488.1149493>.

Zhang, D. and Adipat, B. Challenges, methodologies, and issues in the usability testing of mobile applications. *International Journal of Human-Computer Interaction*, p. 293–308, July 2005.