Evaluating Mobileapp Usability: A Holistic Quality Approach

Luis Olsina[1], Lucas Santos[1], and Philip Lew[2]

[1] GIDIS, Web Engineering School at Universidad Nacional de La Pampa, Argentina
[2] School of Computer Science and Engineering, Beihang University, China
olsinal@ing.unlpam.edu.ar, santos.ls@live.com,
philiplew@gmail.com

Abstract. As newer-generation smartphones enhance functionalities, interactions and services become more complex, leading to usability issues that are increasingly critical and challenging. Also mobile apps have several particular features that pose challenges evaluating their usability using current quality models, usability views, and their relations with target and context entities. With respect to the current literature, usability, actual usability, and user experience are poorly related to target entities (e.g. system and system in use) and context entities, to quality views (e.g. external quality and quality in use), in addition to measurement and evaluation building blocks. In this paper, we propose a holistic quality approach for evaluating usability and user experience of mobile apps. Practical use of our strategy is demonstrated through evaluation for the Facebook mobile app from the system usability viewpoint. Ultimately, a usability evaluation strategy should help designers to understand usability problems effectively and produce better design solutions so we analyze in the context of the framework's applicability toward this goal.

Keywords: Mobile app, Quality model, Usability, User Experience, Evaluation.

1 Introduction

Nowadays, for mobile apps, more robust network infrastructures and smarter mobile devices have led to increased functionality, integration and interactivity thereby warranting special attention in understanding their differences from apps on other platforms from the usability and user experience (UX) point of view because user requirements, expectations, and behavior can be somewhat different with the mobile platform. For instance, the quality design of Operability from a system viewpoint has a much different and greater influence for mobileapp Usability and UX due to the size of the screen and context of the user. Attributes such as button size, placement, color visibility, and widget usage have, for example, a much greater impact on task completion rates and task error rates [1, 5, 16] than for desktop platforms.

Nielsen *et al.* [16] indicate in recent mobile phone studies that usability varies by device category, which is mainly differentiated by screen size such as regular cellphones with small screen; smartphones with midsize screen and full A-Z keypad;

S. Casteleyn, G. Rossi, and M. Winckler (Eds.): ICWE 2014, LNCS 8541, pp. 111–129, 2014.

and full-screen smartphones with a nearly device-sized touch screen. Authors state that regular cellphones "*offer horrible usability, enabling only minimal interaction with websites*" (i.e. mobile webapps); and conclude "*unsurprisingly, the bigger the screen, the better the user experience when accessing websites*". This is supported by authors across several user testing studies from 2009 to 2012, in which the average success rate metric (which measures the percentage of users who were able to accomplish the proposed mobileapp tasks) rated for each mobile device category 44%, 55% and 74% respectively.

Despite these findings, the reader can ask him/herself what do "horrible usability" and "better UX" mean? What is the relationship between Usability and UX? Are they synonym concepts? Evaluating the success rate of users completing tasks correctly (as a performance indicator of effectiveness) is directly related to UX? If users are highly effective in completing tasks but they are unsatisfied due to perceived low app usefulness, then does UX score still high? Does UX depend on app Usability only or also from other characteristics such as Functional and Information Quality, Security, Reliability, and Efficiency? Is UX a quality characteristic of the system (e.g. a mobile app) or of an app in use? And, what about Usability?

Looking for the answers to these questions, we examined the current literature and found that Usability, Actual Usability (in-use) and UX are poorly linked to target entities (e.g. system and system in use) and context entities (e.g. device, environment, user, etc.), in addition to quality views (e.g. external quality and quality in use) and their quality models. Regarding quality models, ISO 25010 [9] outlines a flexible model with product/system quality –also known as internal and external quality (EQ)- and system-in-use quality –also referred to as quality in use (QinU). System quality consists of those characteristics and attributes that can be evaluated with the app in execution state both in testing and in operative stages; while system-in-use quality consists of characteristics and attributes as evaluated by end users when actually executing app tasks in a real context of use. ISO 25010 also delineates a relationship between the two quality views whereby system quality ‘influences’ system-in-use quality and system-in-use quality is determined by (‘depends on’) system quality. Usability is a system quality characteristic, while Effectiveness, Efficiency and Satisfaction are QinU characteristics. However, Actual Usability and UX, as experienced by the end user are missing concepts in the quoted standard.

From the QinU viewpoint, Hassenzahl [7] characterizes a user’s goals into pragmatic, do goals and hedonic, be goals and categorizes system-in-use quality to be perceived in two dimensions, pragmatic and hedonic. Pragmatic quality refers to the system's perceived ability to support the achievement of tasks and focuses on the system’s actual usability in completing tasks that are the ‘do-goals’ of the user. Hedonic quality refers to the system's perceived ability to support the user’s achievement of ‘be-goals’, such as being happy, or satisfied with a focus on self.

Based on ISO 25010 among other works, such as [2, 7], we have developed 2Q2U (*Quality*, *Quality in use*, *actual Usability* and *User experience*) v2.0 [17], which ties together all of these quality concepts by relating system quality characteristics with Actual Usability and UX. Using the 2Q2U quality framework and a tailored strategy, evaluators can instantiate the quality characteristics to evaluate and conduct a systematic evaluation using the ‘depends’ and ‘influences’ relationships [14]. Besides in [12], we have addressed relevant features of mobile apps with regard to Usability and UX in the light of 2Q2U v2.0 quality models, but a global scheme which links main relationships among mobile target and context entities, quality views, characteristics and measurable properties were left for future endeavors.

Therefore, the major contributions of this research are: i) Represent relevant Usability and UX features of mobile apps with regard to system, system-in-use and context entities; ii) Analyze Usability and UX relationships, as well characteristics and attributes for mobile apps in the light of a conceptual framework and evaluation strategy; and iii) Illustrate an evaluation study for Facebook mobile app from the system usability viewpoint, showing the potential positive impact in designing quality interfaces. Lastly, we hope most of the above raised issues will be answered after reading this work.

Following this introduction, Section 2 describes a global scheme which links main relationships among mobile target and context entities, quality views, characteristics and measurable properties, measurement, and evaluation building blocks, with a focus on Usability and UX. Section 3, outlines our conceptual framework and evaluation approach which give support to the above building blocks. Section 4 demonstrates the practical use of our quality framework and evaluation strategy through the Facebook's mobileapp usability case study. Section 5 describes related work and, finally, Section 6 draws our main conclusions and outlines future work.

2 Featuring Mobileapp Usability and UX

For mobile phones, Usability and UX become crucial because users interact with apps –both native mobile apps and mobile webapps- in different contexts using devices with reduced display real estate. In particular the user’s activity at the time of usage, location, and daytime, amongst other influencing factors such as user profile and network performance have an actual impact on the quality of the user’s experience.

This section examines several features relevant for understanding and evaluating Usability and UX for mobile apps. To do this, Fig. 1 depicts the main building blocks which link some relationships among: i) entity categories, quality views/characteristics, and measurable properties (green/orange boxes); ii) measurement (light-blue box); and iii) evaluation (pink box). Next, we examine particularly i) features which allow better understanding non-functional requirements to further specify Usability and UX attributes for interface-, task-, and perception-based evaluations. First, we give a summary followed by deeper discussion.

The 'entity' label in the upper box represents the potential target entity category to be evaluated. It is defined as the "*object category that is to be characterized by measuring its attributes*", while an attribute is "*a measurable physical or abstract property of an entity category*" [18]. There are two instances of (super) categories for target entities that are of interest for evaluating Usability and UX, viz. product/system and system in use. In turn, an entity category can aggregate sub-entities, e.g., a mobile app is composed of basic and advanced GUI objects from the interface standpoint, as shown in Fig. 1. Moreover, lower level entities can be identified for GUI objects like button, menu, widget, etc. Another label in the upper box is 'context' which is defined as "*a special kind of entity representing the state of the situation of a target entity category, which is relevant for a particular information need*". So, system in use is characterized by a context-in-use entity (upper-right orange box) which in turn can aggregate environment, user and task contextual sub-entities.

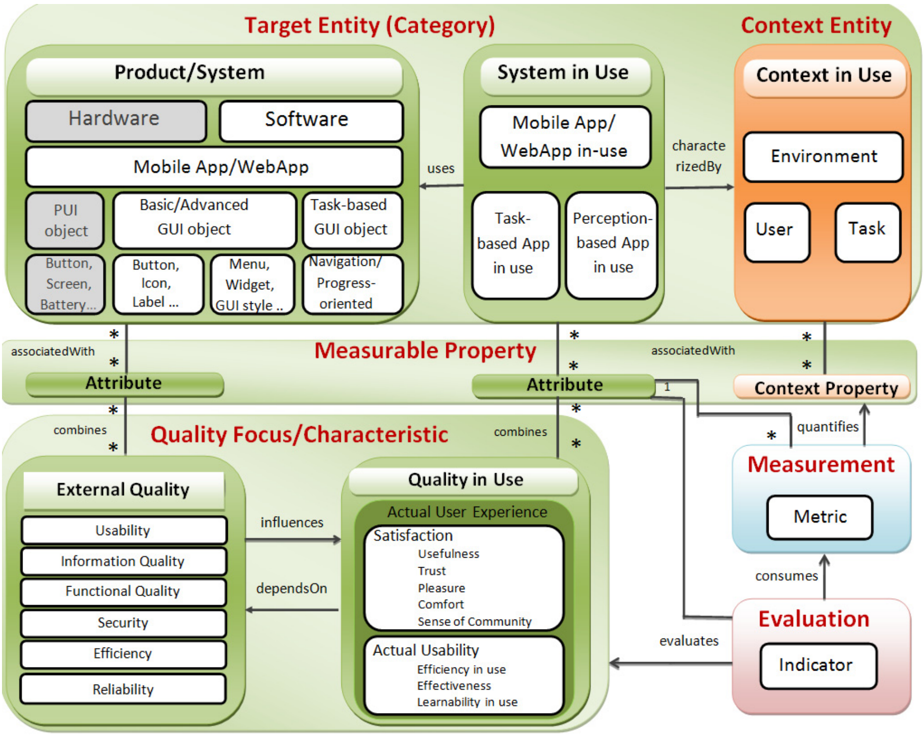

Fig. 1. Global scheme which links the main relationships among mobile Target and Context Entities, Quality Focuses, Measurable Properties, Measurement, and Evaluation building blocks. Note that the two lower-level Product/System sub-entity categories are just addressed for the Usability characteristic, which deals with user interface-oriented evaluation issues –PUI/ GUI stands for Physical/Graphical User Interface.

On the other hand, an entity (category) and their sub-entities cannot be measured and evaluated directly but only by means of the associated measurable properties, i.e. attributes and context properties accordingly (see Fig. 1). Quality models can be the focus for different entities, and usually specify product/system or system-in-use quality requirements regarding main characteristics that can be further subdivided into sub-characteristics, which combine attributes. Product/system quality requirements are modeled by the EQ focus (view), which includes higher-level characteristics such as Usability, Security, Functional Quality, etc. Instead, system-in-use quality requirements are modeled by the QinU view, which include higher-level characteristics such as Actual UX, Satisfaction and Actual Usability.

Lastly, looking at the entity building block relationships, we see the ‘uses’ and ‘characterized by’ relations. Also between EQ and QinU views, we observe that system quality ‘influences’ system-in-use quality or system-in-use quality ‘depends on’ system quality. Note that for instantiated EQ and QinU models these relationships can be explored in light of concrete entity attributes by performing evaluations. For instance, using an evaluation strategy we can explore relationships between system quality and system-in-use quality attributes that may contribute to usage improvements. Regarding the above global scheme, in the next three sub-sections we closely examine the features of mobileapp Usability, UX, and context.

2.1 Featuring Mobileapp Usability

Usability is a characteristic for a system from the EQ viewpoint. It is one out of eight EQ characteristics in 2Q2U v2.0 (see [17] for quality models details). We define Usability as the "*degree to which the product or system has attributes that enable it to be understood, learned, operated, error protected, attractive and accessible to the user, when used under specified conditions*". (Note this definition is very close to that in ISO 9126-1 [11] rather than to [9], as we discuss in related work).

Examining the first part of the above definition, products are entities at early phases of a software life cycle (e.g., textual or graphical documents, etc.); while systems are executable software products (e.g. a mobile app in a testing or operative stage), which could include hardware and software together. Examining the second part of the above definition, we observe that the system (particularly, interface-related objects of the app) has attributes that enable the user to interact considering certain factors. These are the Usability sub-characteristics which can be evaluated through Understandability, Learnability, Operability, User error protection, UI aesthetics and Accessibility. Table 1 shows the Usability sub-characteristics and attributes definitions used in the Facebook evaluation study, in Section 4.

Recalling that characteristics and sub-characteristics combine attributes which are associated to entities (see Fig. 1) some typical mobileapp sub-entities that should be considered for Usability design and evaluation are entry fields and widgets, menus, carousels, breadcrumb path, amongst others. Entity sub-categories specific for Usability evaluation can be physical and graphical user interface (PUI/GUI) objects [6], and task-based GUI objects. A possible categorization for GUI objects can be basic or advanced objects (similar to that described in [8]).

All definitions for sub-characteristics and attributes in Table 1 include to a great extent the referred target sub-entity. For instance, the Visibility (1.3.2 coded) sub-characteristic is defined as "*degree to which the application enables ease of operation through controls and text which can be seen and discerned by the user in order to take appropriate actions*", and one combined attribute viz. Brightness difference appropriateness (1.3.2.1.1) is defined as "*degree to which the foreground color of the GUI object (e.g. text, control, etc.) compared to the background color provide appropriate brightness difference*". Actually, many attributes can determine whether or not the application is easily visible to the user. Depending on the context, different text colors and backgrounds can have a positive or negative impact. Remember that mobile users want to glance quickly and understand and operate almost immediately and there may be glare on their screen if they are outdoors. Also this means that appropriate usage of control/text colors (and size) can greatly impact the user's speed of comprehension and therefore, operational effectiveness and efficiency.

In addition to Usability, characteristics to evaluate other mobileapp EQ aspects are Security, Functional and Information Quality, etc. in which the target sub-entities should be defined accordingly, at least to the system lower layers.

Table 1. Definition of EQ/Usability sub-characteristics and attributes –in *italic*

Characteristic/*Attribute*	**2Q2U v2.0 Definition**
1 Usability	Degree to which the product or system has attributes that enable it to be understood, learned, operated, error protected, attractive and accessible to the user, when used under specified conditions.
1.1 Understandability (synonym Appropriateness Recognizability)	Degree to which users can recognize whether a product or system is appropriate for their needs. Note: Same ISO 25010 definition.
1.1.1 Familiarity	Degree to which the user understand what the application, system's functions or tasks are about, and their functionality almost instantly, mainly from initial impressions
1.1.1.1 Global organization scheme understandability	Degree to which the application scheme or layout is consistent and adheres to either de facto or industry standard to enable users to instantly understand its function and content.
1.1.1.2 Control icon ease to be recognized	Degree to which the representation of the control icon follows or adheres to an international standard or agreed convention.
1.1.1.2.1 Main control icon ease to be recognized	Degree to which the representation of the main controls icons follows or adheres to an international standard or agreed convention.
1.1.1.2.2 Contextual control icon ease to be recognized	Degree to which the representation of the contextual controls icons follows or adheres to an international standard or agreed convention.
1.1.1.3 Foreign language support	Degree to which the application functions, controls and content has multi-language support enabling user to change his/her language of preference.
1.2 Learnability	Degree to which the product or system enables users to learn its app.
1.2.1 Feedback Suitability	Degree to which mechanisms and information regarding the success, failure or awareness of actions is provided to users to help them interact with the application.
1.2.1.1 Current location feedback appropriateness	Degree to which users are made aware of where they are at the current location by an appropriate mechanism.
1.2.1.2 Alert notification feedback appropriateness	Degree to which users are made aware of new triggered alerts that they are involved by an appropriate mechanism.
1.2.1.3 Error message appropriateness	Degree to which meaningful error messages are provided upon invalid operation so that users know what they did wrong, what information was missing, or what other options are available.
1.2.2 Helpfulness	Degree to which the software product provides help that is easy to find, comprehensive and effective when users need assistance
1.2.2.1 Context-sensitive help appropriateness	Degree to which the application provides context sensitive help depending on the user profile and goal, and current interaction.
1.2.2.2 First-time visitor help appropriateness	Degree to which the application provides an appropriate mechanism (e.g. a guided tour, etc) to help beginner users to understand the main tasks that they can do.
1.3 Operability	Degree to which a product or system has attributes that make it easy to operate and control. Note: Same ISO 25010 definition
1.3.1 Data Entry Ease	Degree to which mechanisms are provided which make entering data as easy and as accurate as possible.

Table 1. (*Continued*)

1.3.1.1 Defaults	Degree to which the application provides support for default data.
1.3.1.2 Mandatory entry	Degree to which the application provides support for mandatory data entry.
1.3.1.3 Widget entry appropriateness	Degree to which the application provides the appropriate type of entry mechanism in order to reduce the effort required.
1.3.2 Visibility (synonym Optical Legibility)	Degree to which the application enables ease of operation through controls and text that can be seen and discerned by the user in order to take appropriate actions.
1.3.2.1 Color visibility appropriateness	Degree to which the main GUI object (e.g. text, control, etc.) color compared to the background color provide sufficient contrast and ultimately appropriate visibility.
1.3.2.1.1 Brightness difference appropriateness	Degree to which the foreground color of the GUI object (e.g. text, control, etc.) compared to the background color provide appropriate brightness difference.
1.3.2.1.2 Color difference appropriateness	Degree to which the foreground text or control color compared to the background color provide appropriate color difference.
1.3.2.2 GUI object size appropriateness	Degree to which the size of GUI objects (e.g. text, buttons, and controls in general) are appropriate in order to enable users to easily identify and operate them.
1.3.2.2.1 Control (widget) size appropriateness	Degree to which the size of GUI controls are appropriate in order to enable users to easily identify and operate them.
1.3.2.2.2 Text size appropriateness	Degree to which text sizes and font types are appropriate to enable users to easily determine and understand their meaning.
1.3.3 Consistency	Degree to which users can operate the task controls and actions in a consistent and coherent way even in different contexts and platforms.
1.3.3.1 Permanence of controls	Degree to which main and contextual controls are consistently available for users in all appropriate screens or pages.
1.3.3.1.1 Permanence of main controls	Degree to which main controls are consistently available for users in all appropriate screens or pages.
1.3.3.1.2 Permanence of contextual controls	Degree to which contextual controls are consistently available for users in all appropriate screens or pages.
1.3.3.2 Stability of controls	Degree to which main controls are in the same location (placement) and order in all appropriate screens.
1.4 User Error Protection	Degree to which a product or system protects and prevents users against making errors and provides support to error tolerance.
1.4.1 Error Management	Degree to which users can avoid and recover from errors easily.
1.4.1.1 Error prevention	Degree to which mechanisms are provided to prevent mistakes.
1.4.1.2 Error recovery	Degree to which the application provides support for error recovery.
1.5 UI Aesthetics (synonym Attractiveness)	Degree to which the UI enables pleasing and satisfying interaction for the user. Note: Same ISO 25010 definition.
1.5.1 UI Style Uniformity	Degree to which the UI provides consistency in style and meaning.
1.5.1.1 Text color style uniformity	Degree to which text colors are used consistently throughout the UI with the same meaning and purpose.
1.5.1.2 Aesthetic harmony	Degree to which the UI shows and maintains an aesthetic harmony regarding the usage and combination of colors, texts, images, controls and layouts throughout the whole application.

Table 2. Definition of QinU (sub-)characteristics absent in [9] or were rephrased in 2Q2U v2.0

Characteristic/Sub-characteristic Definition	ISO 25010 QinU Definition
Actual User Experience: Degree to which a system in use enable specified users to meet their needs to achieve specific goals with satisfaction, actual usability, and freedom from risk in specified contexts of use	Note: Absent characteristic in ISO 25010, but similar definition to QinU in this standard
Actual Usability (synonym Usability in use)*:*Degree to which specified users can achieve specified goals with effectiveness, efficiency, learnability in use, and without communicability breakdowns in specified contexts of use	Note: Absent characteristic, but similar concept (i.e. *usability in use*) was in the ISO 25010 draft, and in [2]
Effectiveness: Degree to which specified users can achieve specified goals with accuracy and completeness in specified contexts of use	*Effectiveness*: Accuracy and completeness with which users achieve specified goals
Efficiency (**in use**): Degree to which specified users expend appropriate amounts of resources in relation to the effectiveness achieved in specified contexts of use	*Efficiency:* Resources expended in relation to the accuracy and completeness with which users achieve goals
Learnability (**in use**): Degree to which specified users can learn efficiently and effectively while achieving specified goals in specified contexts of use	Note: Absent characteristic
Sense of Community: Degree to which a user is satisfied when meeting, collaborating and communicating with other users with similar interest and needs	Note: Absent characteristic

As we depict in the next sub-section, UX is a broader concept that depends not only on Usability but also on other system characteristics such as Functional and Information Quality, Security, Reliability, Efficiency, and contexts of use as well.

2.2 Featuring Mobileapp UX

Fig. 1 shows UX as the higher-level characteristic for QinU evaluations. The QinU view characterizes the impact that the system in use (e.g. a mobile app) has on actual users in real contexts of use, i.e., while users perform application tasks in a real environment. Actual UX is defined in Table 2, as "*degree to which a system in use enable specified users to meet their needs to achieve specific goals with satisfaction, actual usability, and freedom from risk in specified contexts of use*".

UX is determined by the satisfaction of the user's be goals (hedonic), and do goals (pragmatic) as noted by Hassenzahl [7]. Moreover, do-goals relate to the user being able to accomplish what they want with Effectiveness and Efficiency (i.e. Actual Usability or Usability in use), while be-goals relate to the user's satisfaction. Satisfaction in [9] includes those subjective, perception-oriented sub-characteristics including Usefulness, Trust, Pleasure, and Comfort -also Sense of Community in [17].

Ultimately, Usability deals with the specification and evaluation of interface-based sub-characteristics and attributes of a system, while Actual Usability deals with the specification and evaluation of task-based sub-characteristics and attributes of an app in use, and Satisfaction with perception-based sub-characteristics and attributes. Recalling that sub-characteristics combine attributes which are associated to entities, we have considered in Fig. 1 two typical app-in-use target sub-entities, namely: Task-based and Perception-based App in-use. The Task-based App-in-use sub-entity can be evaluated using Effectiveness, Efficiency and Learnability in-use attributes. In [14], for the JIRA webapp in-use, we evaluated the "Entering a new defect" task performed by 50 beginner tester users, in which for example Effectiveness combined three attributes such as Sub-task correctness, Sub-task completeness and Task successfulness. (Note that Task successfulness attribute was measured in a similar way that the Average success rate used by Nielsen *et al.* [16]). On the other hand, the Perception-based App-in-use sub-entity can be measured and evaluated using Satisfaction sub-characteristics and attributes that can be included in questionnaire items as in [13], or evaluated through other methods such as observation.

As a final remark, mobileapp selected tasks should be evaluated with respect to real users performing real tasks. This issue includes several key design concerns that have significant impact on the Effectiveness and Efficiency of the final user. For example, *task workflows* need to be designed with the most common tasks in mind that would be suited to mobile usage. Because of the context of use of a mobile user, and the mobile user's limited attention span, the choice of tasks, task workflow and length are extremely important for this limited task set. If task workflows are not designed to be short, there is a higher probability of user error and a lower rate of completion –see Effectiveness definition in Table 2. Workflows therefore need to be compressed by combining several steps into one through careful task definition, evaluation and analysis. Reduced workflows, in turn, reduce task times and increase Efficiency (see definition in Table 2) while, at the same time, reducing error rates and error rate reduction is extremely critical for users with short attention spans. If you are driving and executing a task and get an error, do you continue trying?

2.3 Featuring Mobileapp Context

As mentioned above, the Context entity (category) is a special kind of entity representing the state of the situation of a target entity to be assessed, which is relevant for a particular measurement and evaluation (M&E) information need. Context for a given QinU M&E project is particularly important –i.e. a must-, as instantiation of QinU requirements must be done consistently and in the same context so that evaluations and improvements can be accurately assessed and compared. But also context is important regarding the EQ view, as we describe in Section 4. (Note that in order to reduce clutter in Fig 1, we did not draw an upper-left orange box for product/system context). For instance, system in use is characterized by a context-in-use entity, which in turn can aggregate environment, user and task sub-entities, while a product/system context (for idem target entity), can be characterized by sub-entities such as device (hardware), software, etc.

As commented in [12], the context of a mobileapp user is much different than a traditional desktop webapp user not only due to the size of the screen but also to other situations that influence the user's environment and therefore its behavior. In particular the user's activity at the time of usage, location, amongst other influencing factors such as user profile have actual impact on the quality of the user's experience.

Some a few of these factors considering context sub-entities and properties include: i) *Activity*: What users are doing at the time of usage have a significant influence on the user's attention span; e.g. if they are driving, then they have a very short attention span, maybe 1 second, versus if they are in the middle of a conversation, perhaps they have an attention span of 3 seconds; ii) *Day/time of day*: The day and time can impact what a user is doing, and the level of natural light. Unlike desktop apps which are typically accessed indoors, the usage of mobile apps is particularly sensitive to this contextual factor influencing visibility; ii) *Location*: The location of the user influences many elements; e.g. indoors, outdoors, in a car or in an elevator, all of which can also be related to the user activity; iii) *Network performance*: Obviously the speed at which an app uploads and downloads data is going to have a great impact because of the decreased attention span; iv) *User profile*: The increasing complexity of software combined with an aging user demographic has an interesting effect on the usability of mobile apps. For aging users, usually their close range vision capability has diminished along with their dexterity. On the other hand, apps have become complex, and therefore function and content simplicity and understandability are also critical and influenced by the particular user group. Not only are there more aging users, there are also more younger users as children these days begin using computing devices as toddlers; v) *Device*: The size and type of the device and its physical features influence what the user can see (or not see) as well as the placement and number of controls and widgets in reduced real-estate displays.

This shortage of resources and particular contexts of use all impact on the UX. Lastly, context properties are not part of the EQ or QinU models, but should be recorded accordingly for characterizing the situation of the target entities at hand.

3 Conceptual Framework and Evaluation Approach

3.1 M&E Conceptual Framework

At this point, it is worth mentioning that the main building blocks depicted in Fig. 1 are grounded in a M&E conceptual framework. We have built –as part of evaluation strategies- the C-INCAMI (*Contextual-Information Need*, *Concept model*, *Attribute*, *Metric* and *Indicator*) conceptual framework [18], which is structured in six components, namely: (a) Measurement and Evaluation Project; (b) Non-functional Requirements; (c) Context; (d) Measurement; (e) Evaluation; and (f) Analysis and Recommendation. Each component contains key terms and relationships. Fig 2 shows, for illustration purpose, the (b), (c), and (d) components whose colors match those green/orange/light-blue boxes of Fig. 1.

In fact, the different labels in Fig 1 are mostly instances of the concepts, properties, and relationships included in the C-INCAMI conceptual framework. For instance "System" and "System in Use" in Fig. 1 are two instances of the *Entity Category* term; specifically, each string is the value of the *name* field in Fig. 2. Since an entity category can have *sub-entity* categories, "Basic/Advanced GUI object", "Menu", etc. are instances of sub-entity categories. *Entity* term represents a concrete object; for example, "Facebook mobile app" is the entity *name* that *belongs to* the "System" category regarding the EQ *focus*.

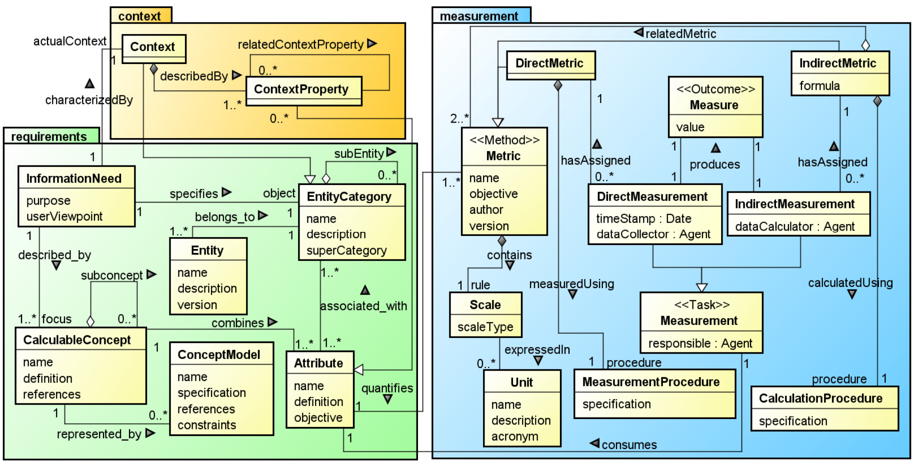

Fig. 2. C-INCAMI Nonfunctional Requirements, Context, and Measurement components

Therefore, the *requirements* component specifies the *Information Need* for a M&E project, i.e., the *purpose* (e.g. "understand", "improve") and the *user Viewpoint* (e.g. "final user", "developer"). In turn, it *focuses* on a *Calculable Concept* (i.e. characteristics whose *names* are for example "External Quality", "User Experience", etc.) and *specifies* the *Entity Category* to be evaluated. On the other hand, a calculable concept and its *sub-concepts* (e.g. "Usability") can be *represented by* a *Concept Model* (e.g. an "EQ model") where the leaves of an instantiated quality model are *Attributes* which are *associated with* a target entity. Table 1 specifies the requirements tree for "Usability", which contains the *names* and *definitions* for the selected sub-characteristics and attributes used in the Facebook mobileapp evaluation.

The *context* component (in Fig. 2) shows explicitly that *Context* is a special kind of *Entity Category*. Context represents the state of the situation of a target entity, which is relevant for a particular information need. To describe the context sub-entities (e.g. "Environment", "Device", etc.) *Context Properties* are used, which are also attributes. Additionally, attributes –as measurable properties- can be *quantified* by metrics and interpreted by indicators.

Metric is a key term in the *measurement* component in Fig 2 (see also Fig. 1). This component allows specifying *Direct* and *Indirect Metrics* used by *Direct* and *Indirect Measurement* tasks which produce *Measures.* A metric is "*the defined measurement or calculation procedure and the scale*". So a metric represents the *how*, that is to say, the method that should be assigned to the steps of a measurement task (the *what*). Lastly, a *Measure* is the number or category assigned to an attribute by making a measurement upon a concrete entity. In order to illustrate the added value of a well-defined measurement component, Table 3 shows the derived template for indirect and direct metric specifications to the "Permanence of main controls" attribute. The "Operability" sub-characteristic combines this attribute that is coded 1.3.3.1.1 in Table 1. Additionally, the screenshot in Fig. 3.b shows the concrete sub-entity named "Main controls bar" that can be further measured.

Table 3. Indirect and direct metric specifications to the *Permanence of main controls* attribute

Target Entity Category: **Name**: *System*; **Sub-Entity Category: Name**: *Smartphone mobile app*; **Concrete Entity:** **Name**: *Facebook app*; **Version**: 3.8; **Sub-Entity Description:** Set of *Screens* of the *Facebook app* where the *Main controls bar* is (or should be) containing the set of *Main controls (Buttons)* **Attribute:** **Name**: *Permanence of main controls*; **Code**: *1.3.3.1.1* in Table 1 **Definition**: Degree to which main controls are consistently available for users in all appropriate screens or pages; **Objective**: To determine the degree to which the main controls are present in all appropriate screens. **Indirect Metric:** **Name**: *Ratio of Main Controls Permanence* (%MCP); **Objective**: To determine the percentage of permanence for controls from the set of main controls in the application selected screens; **Author**: Santos L.; **Version**: 1.0; **Calculation Procedure: Formula**: $\%\mathrm{MCP} = \left[\frac{\sum_{i=1}^{m}\sum_{j=1}^{n}\mathrm{MCPLij}}{(m*n)}\right] * 100$; for i=1 to m and j=1 to n, where m is the number of application main controls and n is the number of application selected screens; with m, n > 0 **Numerical Scale: Representation**: Continuous; **Value Type**: Real; **Scale Type**: Ratio; **Unit Name**: Percentage; **Acronym**: % **Related Metrics**: Main control permanence level (MCPL) **Related Direct Metric:** **Name**: *Main Control Permanence Level* (MCPL); **Objective:** To determine the permanence level of a selected control in a given application screen; **Author**: Santos L.; **Version**: 1.0; **Measurement Procedure: Type**: Objective; **Specification**: The expert inspects the main controls bar in a given screen in order to determine whether the button is available or not, using the 0 or 1 allowed values. Where 0 means the main button is absent in the screen, and 1 means the main button is present in the screen; **Numerical Scale: Representation**: Discrete; **Value Type**: Integer; **Scale Type**: Absolute; **Unit: Name** Control

3.2 Evaluation Approach and Strategies

This sub-section gives a summary of our generic evaluation approach, which is made up of a *quality modeling framework* and *M&E strategies*, where a concrete strategy should be selected for purposefully instantiating quality models, processes, and performing evaluations for a concrete project information need. Particularly, the generic evaluation approach relies on two pillars, namely: i) a *quality modeling framework* –where 2Q2U v2.0 is a subset [17], which includes the EQ and QinU views and the 'depends' and 'influences' relationships between them; and ii) *M&E strategies*, which in turn are based on three principles viz. a *M&E conceptual framework* (as introduced in the previous sub-section), *process view specifications*, and *method specifications*.

So far, we have developed two integrated strategies which include these three principles, namely: GOCAME (*Goal-Oriented Context-Aware Measurement and Evaluation*) [17, 18], and SIQinU (*Strategy for understanding and Improving Quality in Use*) [14]. GOCAME is a multi-purpose strategy that follows a goal-oriented and context-based approach in defining and performing M&E projects. GOCAME is a multi-purpose strategy because it can be used to evaluate (e.g. "understand", "improve", etc.) quality not only for product, system and system-in-use entities but also for others such as resource, by using their instantiated quality models and tailored process accordingly. However, GOCAME does not incorporate the QinU/EQ/QinU relationships and improvement cycles as in SIQinU. Rather it can be used to understand the current or future situation, as an evaluation snapshot, of concrete entities. On the other hand, SIQinU is a specific-purpose strategy, which has specific processes, methods and change procedures that are not specified in GOCAME. Ultimately, given the target information need and objective, we can select the specific strategy and its tailored processes and methods in order to fulfill that specific goal.

For example, GOCAME has a well-defined M&E process specification, which is composed of six generic activities, namely: (A1) *Define Non-functional Requirements*; (A2) *Design the Measurement*; (A3) *Implement the Measurement*; (A4) *Design the Evaluation*; (A5) *Implement the Evaluation*; and (A6) *Analyze and Recommend*. Each activity can be accordingly tailored for a specific quality focus regarding the information need, e.g. if the focus is on EQ then A1 is named *Define Non-functional Requirements for EQ*, and so on. Instead, if the focus is on QinU then A1 is named *Define Non-functional Requirements for QinU*, and so forth. Note that in our process specifications each activity is not atomic, so it should be decomposed into tasks.

Lastly, the strategies' activities are supported by different method specifications. Since the M&E strategies rely on the quality modeling framework which is made up of quality models, inspection of characteristics and attributes is the basic method category. Attributes are supported by metric and elementary indicator method specifications, while quality models are calculated using different indicator aggregation methods such as LSP (*Logic Scoring of Preference*) [4], which is a weighted multi-criteria aggregation method. However, user testing and inquiry method categories can be used –mainly for QinU- meanwhile attributes of Efficiency, Effectiveness, Learnability in use and Satisfaction can be derived from task usage log files, questionnaire items, etc., as we did in [13, 14]. For planning and performing changes traditional methods and techniques such as refactoring, re-structuring, re-parameterization, among others can be used as well. The next section demonstrates a practical use of our quality framework and GOCAME strategy through excerpts of our Facebook's mobileapp Usability evaluation study.

4 Usability Evaluation for the Facebook Mobile App

The abovementioned A1 activity named *Define Non-functional Requirements for EQ* has a specific goal or problem as input and a nonfunctional specification document as output. A1 consists of: *Establish EQ Information Need* (A1.1), *Select an EQ Model* (A1.2), and *Specify (System) Context* (A1.3) sub-activities [17].

Considering A1.1, the *purpose* of the information need is to "understand" the current EQ satisfaction level achieved, particularly by evaluating the "Usability" strengths and weaknesses from the "final user" *viewpoint.* "Facebook mobile app" is the concrete entity whose sub-entities for the Usability *focus* are related to basic-, advanced- and task-based GUI objects (recall Fig. 1). For example, in the Fig. 3.b screenshot the "Main controls bar" contains a set of main controls or "Buttons".

For the given focus, the A1.2 sub-activity allows selecting from a repository the sub-characteristics and attributes to be included. Table 1 documents the resulting requirements tree, which includes "Understandability" (1.1), "Learnability" (1.2), "Operability" (1.3), "User error protection" (1.4), and "Aesthetics" (1.5) sub-characteristics. For example, "Operability" includes in turn sub-characteristics such as "Visibility" (1.3.2), "Consistency" (1.3.3), etc., which *combine* attributes *associated* to the entities. Particularly, "Color visibility appropriateness" (1.3.2.1) combines two attributes associated to "Color/Text" objects (see Fig.3.a); while "Permanence of controls" (1.3.3.1) combines two attributes associated to "Main controls bar" and "Contextual control" objects (Fig.3.a and b). For instance, the "Permanence of main controls" attribute (1.3.3.1.1) is defined in Table 1 as "*degree to which main controls are consistently available for users in all appropriate screens or pages*". Note that in the main controls bar of the three shown screenshots all buttons are not always available, so the measured value will be produced in the A3 (*Implement the EQ Measurement*) activity, using the appropriate metric from Table 3.

Fig. 3. Three Facebook screenshots: a) Contextual control and Color/Text objects are highlighted; b) The Main controls bar, and the chat Button, which is not available in the other two screenshots; and c) A typical date widget used when create event task is performed

Lastly, the A1.3 sub-activity deals with the selection of *context properties* (and further values) like "mobile device type" (e.g. tablet, mobile phone); "mobilephone generation" (e.g. regular cellphone, mid-sized smartphone, full-sized smartphone [16]); "mobilephone device brand-model"; "target mobileapp type" (e.g. native mobile app, mobile webapp), among many others, recalling that *context* is a special kind of entity related to the entity category to be evaluated, as mentioned in sub-section 2.3.

Once the information need, EQ requirements, and context specifications were yielded, the next A2 activity, *Design the EQ Measurement*, consists of selecting the meaningful metrics from a repository to quantify the 23 measurable attributes. One direct or indirect metric should be assigned per each attribute of the requirements tree respectively. For example, the "Ratio of Main Controls Permanence" (%MCP) *indirect metric* whose *objective* is "*to determine the percentage of permanence for controls from the set of main controls in the application selected screens*" was chosen to quantify the 1.3.3.1.1 attribute, as shown in Table 3. While an indirect metric has a *calculation procedure* for its *formula* specification, a direct metric has a *measurement procedure*. %MCP includes a related direct metric where the measurement procedure indicates "*the expert inspects the main controls bar in a given screen in order to determine whether the button is available or not, using the 0 or 1 allowed values. Where 0 means the main button is absent in the screen, and 1 means the main button is present in the screen*". In summary, 48 metrics were designed for this study taking into account direct, indirect and related metrics for the latter.

The A3 activity produces the *measures* for all metrics at given moments in time as well the linked data to concrete object references and parameters. Data collection for metrics on the Facebook app (ver.3.8 for Android) were performed from Dec 26-28, 2013. The measure for %MCP gave 54.9% of permanence of main controls, regarding that 5 main buttons should be placed in the main controls bar in 35 appropriate screens (out of 38 app screens). Looking at Fig 3, we can observe for example that the "chat" button is absent in screens a) and c), so if the end user wants to trigger this action on those screens, he/she needs to perform more clicks than needed to initiate the task.

Table 4. Excerpt of *Usability* sub-characteristics and attributes from Table 1. Only *Operability* sub-characteristics and attributes are fully shown with Elementary Indicator values (2nd column); the 3rd column shows Partial/Global Indicator values, which are all in % scale unit.

1 Usability		**60.5**
1.1 Understandability		**76.1**
1.2 Learnability		**59.7**
1.3 Operability		**80.7**
1.3.1 Data Entry Ease		90
1.3.1.1 Defaults	100	
1.3.1.2 Mandatory entry	50	
1.3.1.3 Widget appropriateness	100	
1.3.2 Visibility (synonym **Optical Legibility**)		81.5
1.3.2.1 Color visibility appropriateness		100
1.3.2.1.1 Brightness difference appropriateness	100	
1.3.2.1.2 Color difference appropriateness	100	
1.3.2.2 GUI object size appropriateness		63
1.3.2.2.1 Control (widget) size appropriateness	100	
1.3.2.2.2 Text size appropriateness	42.1	
1.3.3 Consistency		75.5
1.3.3.1 Permanence of controls		57.3
1.3.3.1.1 Permanence of main controls	54.9	
1.3.3.1.2 Permanence of contextual controls	67.4	
1.3.3.2 Stability of controls	95.5	
1.4 User Error Protection		**8.4**
1.5 UI Aesthetics		**80.8**

Once metrics were selected for quantifying all attributes, then A4 can be performed, which deals with designing the EQ evaluation. For space reasons, we did not describe the *evaluation* component in sub-section 3.1, but a key concept is *Indicator*, as shown in Fig. 1. While an *elementary indicator* evaluates the satisfaction level met for an elementary requirement, i.e., an attribute of the requirements tree, a *partial/global indicator* evaluates the satisfaction level achieved for partial (sub-characteristic) and global (characteristic) requirements represented in the quality model. Therefore, a new *scale* transformation and *decision criteria* (in terms of *acceptability levels* and ranges) are defined. In this study, we used three acceptability ranges in a percentage scale: a value within 60-80 (a marginal –yellow- range) indicates a need for improvement actions; a value within 0-60 (an unsatisfactory –red- range) means change actions must take place with high priority; and a score within 80-100 indicates a satisfactory level –green- for the analyzed attribute or characteristic.

Details of elementary and global evaluation, as well as the LSP model used in this study to calculate indicators (A5 activity) can be referred elsewhere [18]. Table 4 shows the elementary and partial indicators' values for "Operability", and only partial and global indicators' values for the other sub-characteristics in addition to the acceptability levels achieved.

Finally, GOCAME projects record all data, metadata and information coming from metrics and indicators as well as the quality model and context specifications and values. The *Analyze and Recommend* (A6) activity produces a recommendation document, which can facilitate planning actions for further improvement.

Based on indicator results shown in the 3rd column of Table 4, we can observe that the Usability characteristic in the Facebook app reached a marginal acceptability level (60.5%), which means a need for improvement actions. Taking into account its sub-characteristics viz. Understandability (1.1), Learnability (1.2) and User Error Protection (1.4) reached a marginal and unsatisfactory acceptability levels respectively. Therefore some of their elementary indicators are performing weakly and surely need recommendations for attribute changes.

On the other hand, Operability (1.3) met the satisfactory level of 80.7%. However, this does not imply that there are no weakly performing attributes for Operability. While Color Visibility Appropriateness (1.3.2.1) scored 100 in its two attributes, Permanence of Controls (1.3.3.1) scored in its two attributes, 54.9 and 67.4 respectively (see 2nd column), so a recommendation for further improvement can be made. For understanding the reasons and planning change actions, the metric specification and the measured values are central in GOCAME for these endeavors. %MCP metric allowed (in A3) to store per each main button its availability in each corresponding app screen in which must stay. So evaluators can easily understand, for instance, where the "chat" button is absent, and so for each button of the main controls bar. Also, the tailored strategy may help designers to understand and act on (system) usability problems effectively to produce better design solutions as well.

Finally, as the reader can surmise the metric design specification helps not only planning the change action, but also gauging (predicting) the improvement gain once the action is performed. Ultimately, if we add a new activity to the six described GOCAME activities (as we did it in fact previously with SIQinU), e.g. (A7) *Plan and Perform Improvement Actions*, then the A3, A5 and A6 can be fully reused for re-evaluation and analysis of the improvement gain with regard to the previous app version. Note that changes should be made on the app entity not on the app in use.

5 Related Work and Discussion

As commented previously, in the state-of-the-art literature, Usability, Actual Usability and UX features are very often poorly linked to target entities (e.g. system and system in use) and context entities (e.g. device, environment, user, etc.), in addition to EQ and QinU views and their relationships. Bevan [3] states that international standards for Usability should be more widely used because one of their main purposes is to impose consistency, compatibility, and safety. Usability has also been integrated into standards for software quality and evaluation; e.g. ISO 25010 (which supersedes to ISO 9126-1 [11]) provides a comprehensive structure for the role of Usability as part of system quality as well a broader concept of QinU increasing the business relevance of usability in many situations. Besides, author indicates that referring to the terminology from the field of software quality, it can be said that UX is more related to the concept of QinU, whereas Usability more to EQ.

From our viewpoint, one of the strengths in ISO 25010 is not only the quality models and included characteristics but also the two quality views and relationship whereby system quality 'influences' system-in-use quality and system-in-use quality 'depends on' system quality. However, some weaknesses we point out are: UX is still an absent characteristic in ISO; there exists a dual Usability definition which blurs system Usability with QinU meanings (i.e. Usability is defined in [9] 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*", while QinU as "*degree to which a product or system can be used by specific users to meet their needs to achieve specific goals with effectiveness, efficiency, freedom from risk and satisfaction in specific contexts of use*") so, we consider the Usability definition given in [11] (adapted in Table 1) is closer to the intended aim; and, the Context coverage characteristic included in the [9] QinU model, which can be represented independently of quality models, as shown in previous sections and in [17]. Therefore, in order to bridge this gap, we have developed the 2Q2U v2.0 quality modeling framework, considering also contributions such as [2, 7], amongst others.

On the other hand, in the Apple [1] and Google [5] design and user interface guidelines, the relationship between mobileapp entities with Usability and UX concepts is not definitively explicit in models, nor is it represented in the Usability works in [16], nor in other quality-related research such as [15, 19]. For instance, Nielsen *et al.* list out in [16] many features and checklists of mobile apps in that would be desired or needed in certain contexts of use but do not use quality views and modeling approaches. Therefore, the capability for consistent application using a conceptual framework and strategies to systematically apply concepts and evaluate and improve a mobile app is rather limited. (Recall the raised issues in Section 1).

Lastly, a holistic approach similar to ours for evaluating the Usability of mobilephones in an analytical way is documented in [8], which is based on a multi-level, hierarchical model of Usability factors. These factors related to views and entities are collectively measured to give a single score with the use of checklists. Moreover, the conceptual framework and strategy involves a hierarchical model of usability factors, four sets of checklists, a quantification method, and an evaluation process. The conceptual framework for usability indicators [6] is based on the ISO 15939 [10] measurement model, which its terms are structured in a glossary. Conversely, we developed an ontology [18] for the C-INCAMI M&E components (recall Fig. 2) where [10] was one of the used sources. Consequently, from components we derive metric an indicator metadata in templates (as in Table 3) that allows consistency and repetitively among projects and analysis of data. As added value, a well-designed metric helps not only to yield measures but also to plan change actions on the product/system attribute or capability. Finally, the process in the [8] strategy is poorly specified compared to that in GOCAME or SIQinU [14] strategies.

6 Conclusions

As the contributions mentioned in the Introduction Section, firstly, we have characterized and represented relevant Usability and UX features of mobile apps with regard to system, system-in-use and context entities. Secondly, we have analyzed Usability, Actual Usability and UX relationships regarding also EQ and QinU views, as well as specific Usability sub-characteristics and attributes for mobile apps in the light of a holistic evaluation approach. This evaluation approach is made up of a *quality modeling framework* (where 2Q2U is a subset) and *M&E strategies*, which in turn are based on three principles namely: a *M&E conceptual framework* (i.e. the C-INCAMI conceptual framework which is rooted in ontologies), *process view specifications*, and *method specifications*. So given the target information need, we can select the specific strategy and its tailored processes and methods in order to fulfill that specific purpose aimed at performing evaluations, analysis and recommendations. To this, we illustrated an evaluation study for the Facebook mobile app from the system Usability viewpoint, using the GOCAME strategy.

Of course, the Facebook study was made on the basis of a proof of concept as a typical social network app. But if we could have had control of the source code obviously GOCAME can be tailored to support change actions based on recommendations for improvement on weak performing indicators and re-evaluation of the new app version. An app with design features that jeopardize Effectiveness, Efficiency, Safety or Satisfaction (i.e. the do and be UX goals) can potentiate risks that it will not meet its business objectives. Evaluating these high level non-functional requirements such as Satisfaction, Actual Usability may feed back into detailed Usability, Functional and Information Quality, Security, etc. attributes and design requirements to maximize the quality of the user's experience and to minimize the likelihood of adverse consequences. Hence, our holistic evaluation approach can give support by means of specific strategies –as SIQinU- to the QinU/EQ/QinU improvement cycles. Ongoing research focuses on further utilizing our evaluation approach for QinU/EQ/QinU cycles for improving the design of mobile apps.

References

1. Apple iOS Human Interface Guidelines (2014), `http://developer.apple.com/library/ios#documentation/UserExperience/Conceptual/MobileHIG/Introduction/Introduction.html` (retrieved by January)
2. Bevan, N.: Extending Quality in Use to provide a Framework for Usability Measurement. In: Kurosu, M. (ed.) HCD 2009. LNCS, vol. 5619, pp. 13–22. Springer, Heidelberg (2009)

3. Bevan, N.: International Standards for Usability Should Be More Widely Used. Journal of Usability Studies 4(3), 106–113 (2009)
4. Dujmovic, J.: Continuous Preference Logic for System Evaluation. IEEE Transactions on Fuzzy Systems 15(6), 1082–1099 (2007)
5. Google User Interface Guidelines (2014), `http://developer.android.com/guide/practices/ui_guidelines/index.html` (retrieved by January)
6. Ham, D.-H., Heo, J., Fossick, P., Wong, W., Park, S., Song, C., Bradley, M.: Model-based Approaches to Quantifying the Usability of Mobile Phones. In: Jacko, J.A. (ed.) HCI 2007. LNCS, vol. 4551, pp. 288–297. Springer, Heidelberg (2007)
7. Hassenzahl, M.: User Experience: Towards an experiential perspective on product quality. In: 20th Int'l Conference of the Assoc. Francophone d'IHM, vol. 339, pp. 11–15 (2008)
8. Heo, J., Ham, D.-H., Park, S., Song, C., Chul, W.: A framework for evaluating the usability of mobile phones based on multi-level, hierarchical model of usability factors. Interacting with Computers 21(4), 263–275 (2009)
9. ISO/IEC 25010: Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models (2011)
10. ISO/IEC 15939: Software Engineering - Software Measurement Process (2002)
11. ISO/IEC 9126-1: Software Engineering - Product Quality - Part 1: Quality Model (2001)
12. Lew, P., Olsina, L.: Relating User Experience with MobileApp Quality Evaluation and Design. In: Sheng, Q.Z., Kjeldskov, J. (eds.) ICWE Workshops 2013. LNCS, vol. 8295, pp. 253–268. Springer, Heidelberg (2013)
13. Lew, P., Qanber, A.M., Rafique, I., Wang, X., Olsina, L.: Using Web Quality Models and Questionnaires for Web Applications Evaluation. In: IEEE Proc., QUATIC, pp. 20–29 (2012)
14. Lew, P., Olsina, L., Becker, P., Zhang, L.: An Integrated Strategy to Systematically Understand and Manage Quality in Use for Web Applications. Requirements Engineering Journal 17(4), 299–330 (2012)
15. Nayebi, F., Desharnais, J.-M., Abran, A.: The state of the art of mobile application usability evaluation. In: 25th IEEE Canadian Conference on Electrical Computer Engineering, pp. 1–4 (2012)
16. Nielsen, J., Budiu, R.: Mobile Usability. New Riders, Berkeley (2012)
17. Olsina, L., Lew, P., Dieser, A., Rivera, B.: Updating Quality Models for Evaluating New Generation Web Applications. In: Abrahão, S., Cachero, C., Cappiello, C., Matera, M. (eds.) Journal of Web Engineering, Special Issue: Quality in New Generation Web Applications, vol. 11(3), pp. 209–246. Rinton Press, USA (2012)
18. Olsina, L., Papa, F., Molina, H.: How to Measure and Evaluate Web Applications in a Consistent Way. In: Rossi, G., Pastor, O., Schwabe, D., Olsina, L. (eds.) Web Engineering: Modeling and Implementing Web Applications. HCIS, pp. 385–420. Springer (2008)
19. Sohn, T., Li, K.A., Griswold, W.G., Holland, J.: A diary study of mobile information needs. In: ACM, Conference CHI 2008, Florence, Italy, pp. 433–442 (2008)