Towards Understanding and Improving Mobile User Experience

Philip Lew and Luis Olsina
philip.lew@xbosoft.com, olsinal@ing.unlpam.edu.ar

Abstract

Design and evaluation of traditional web applications cannot account for the particular features and usage contexts of mobile applications (MobileApps). MobileApps have several characteristics that pose challenges in their design and evaluation regarding current quality models and their included characteristics and sub-characteristics. For instance, user interface operability has a much different and greater influence when evaluating MobileApp usability and user experience due to the context of the user. Characteristics such as multi-touch gestures, button size, and widget usage have a magnified impact on task completion rates.

In this paper, we propose utilizing our previously developed ISO 25010-based quality models and framework so-called 2Q2U (Quality, Quality in use, actual Usability and User experience) as a basis to understand, evaluate and improve MobileApp user experience. Specific MobileApp task screens and attributes are illustrated in order to show our evaluation approach applicability.

Biography

After working in various management and technical positions in software product development and product management, today, Phil leads XBOSoft’s (www.xbosoft.com) direction and strategy as their CEO. His Ph.D. research in software quality and usability resulted in several IEEE and ACM journal publications and he has been published in various trade journals as well. He has presented at several conferences including Software Test Professionals 2012, and the International Conference of Web Engineering-2009-10-11 on web application usability, user experience, and quality evaluation. In the past 20 years, he has helped hundreds of organizations assess the quality of their software, examine software quality processes and set forth measurement plans so that they can consistently improve software quality using systematic methods. He received his B.S. and Master of Engineering degrees in Operations Research from Cornell University. His current post-doctorate research areas are focused on user experience measurement frameworks and software learnability.

Luis Olsina is Full Professor in the Engineering School at the National University of La Pampa, Argentina, and heads the Software and Web Engineering R&D group. His research interests include software and web engineering, particularly, quality requirements modeling, web quality strategies, quality improvement, measurement and evaluation processes, evaluation methods and tools, and domain ontologies. He earned a PhD in the area of software engineering and a MSE from National University of La Plata, Argentina. In the last 17 years, he has published over 100 refereed papers, and participated in numerous regional and international events both as program committee chair and member. Recently, Luis and his colleagues have co-edited the book titled Web Engineering: Modeling and Implementing Web Applications.
1. Introduction

Web applications (WebApps), have become the most predominant form of software delivery today with users and businesses choosing to rent software rather than buy it. The concept of renting versus buying, and the availability of free trials and ease of switching software has led to increased and focused attention on WebApp quality and software quality in general. Users (customers) cannot easily switch and are not tied in with a large capital expenditure. Software quality models in conjunction with an evaluation strategy facilitate understanding, recommending, and especially improving their quality.

With respect to software quality models, ISO 25010 (ISO/IEC 25010, 2011) 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). Product quality consists of those characteristics that can be evaluated in early development stages, for instance, design documents, code quality, etc., while system quality consists of those characteristics and attributes that can be evaluated in late stages, with the application in execution state. On the other hand, system-in-use quality consists of characteristics as evaluated by an end user when actually executing application tasks in a real context. An example would be a nurse or a doctor entering patient record and diagnosis information into an electronic health records system. A doctor even while doing the same task, will have different error and completion rates than a nurse. Doctors may also take longer and have less efficiency in completing tasks simply because they don’t do as many. In addition, system QinU is heavily dependent on the context of the task and user. For instance, a user in a dim warehouse doing inventory will have a different viewpoint and user experience than a doctor in a well-lit hospital.

ISO 25010 also delineates a relationship between the two quality views whereby system quality ‘influences’ system-in-use quality and system-in-use quality ‘depends’ on system quality. We recently developed 2Q2U version 2.0 (Olsina et. al. 2012), which ties together all of these quality concepts by relating system quality characteristics and attributes with QinU and user experience (UX). By instantiating 2Q2U, evaluators can select the quality characteristics to evaluate and conduct a systematic evaluation using the ‘depends’ and ‘influences’ relationships in conjunction with a strategy (Lew, Olsina et. al. 2012).

Today, for MobileApps, more robust network infrastructures and smart mobile devices have led to increased functionality and capability thereby warranting special attention in comprehending how they are different from the UX point of view because user requirements, expectations, and behavior can be somewhat different. For instance, the quality design and evaluation 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. Characteristics such as button size, placement, contextual help, and widget usage for example have a much greater impact on task completion rates and task error rates (Apple, 2013; Budin et. al. 2013; Google, 2013; Nielsen, 2011; Olsina et al. 2012). Ultimately, UX characteristics are very often neglected in quality modeling or seldom placed appropriately in quality views and this is magnified in a mobile context (Nayebi et. al. 2012; Olsina et al. 2012).

Given this, there is a need for a characterization of MobileApps considering non-functional aspects in both UX and EQ. Consequently, the particular features of MobileApps –regarded both as a system and a system-in-use entity- pose new challenges regarding current quality models and their included and more relevant characteristics and sub-characteristics, as well as the particular attributes or properties to be measured and evaluated, e.g., by metrics and indicators. Additionally, for MobileApps, there is an increased emphasis on many contextual elements related to the task and therefore the QinU. So, starting with the task at hand, and applying our 2Q2U model, we can incorporate the importance of task and particular MobileApp UX factors into the design of MobileApps. Starting with MobileApp UX, we can outline practical guidelines to design a MobileApp with optimal EQ characteristics based on UX goals.

Following this introduction, Section 2 outlines our 2Q2U quality framework for better understanding where UX and usability fit in. Section 3, discusses relevant features of MobileApps (both as a system and a system-in-use entity category) useful for designing and evaluating UX and usability. In Section 4, we discuss the usefulness of the proposed framework, while examples of UX/usability attributes and screens for MobileApps are illustrated. Section 5 draws our main conclusions and outlines future work.
2. 2Q2U Quality Framework

Regarding quality models, we recently enhanced the ISO 25010 external quality and quality in use models, while maintaining many characteristics and the ‘depends’ and ‘influences’ relationships between both views. Figure 1 depicts the quality model enhancement named 2Q2U, which was used and instantiated in different case studies (Lew, Qanber Abbasi et. al. 2012; Olsina et. al. 2012).

For the QinU model we have added two main characteristics which are absent in the ISO standard viz. Actual User Experience, and Actual Usability. These concepts are related hierarchically as shown in Figure 1 and also defined in Table 1.

For the EQ model, we have also rephrased 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”. Ultimately, the rationale of these adaptations is in (Olsina et. al. 2012).

![Figure 1: 2Q2U v2.0 model characteristics with some sub-characteristics and relationships](image-url)
Table 1: Definitions of QinU characteristics and sub-characteristics that are absent in ISO 25010 or were rephrased in 2Q2U v2.0

<table>
<thead>
<tr>
<th>QinU Characteristic or Sub-characteristic</th>
<th>ISO 25010 QinU Definition (ISO/IEC 25010, 2011)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual User Experience</td>
<td>Note: Absent calculable concept</td>
</tr>
<tr>
<td>Actual Usability</td>
<td>Note: Absent calculable concept, but similar concept (i.e. <em>usability in use</em>) was in the ISO 25010 2009 draft</td>
</tr>
<tr>
<td>Effectiveness</td>
<td>Effectiveness: Accuracy and completeness with which users achieve specified goals</td>
</tr>
<tr>
<td>Efficiency</td>
<td>Efficiency: Resources expended in relation to the accuracy and completeness with which users achieve goals</td>
</tr>
<tr>
<td>Learnability in use</td>
<td>Note: Absent calculable concept</td>
</tr>
<tr>
<td>Communicability</td>
<td>Note: Absent calculable concept</td>
</tr>
<tr>
<td>Sense of Community</td>
<td>Note: Absent calculable concept</td>
</tr>
</tbody>
</table>

In using the ISO 25010 quality models for MobileApps, it is assumed that the software quality models, definitions, and concepts in the standard were intended for application to software information systems as a whole and therefore are also applicable to a great extent to MobileApps, a type of software application. Moreover, we have built up 2Q2U on top of ISO standards in order to consider new features of new generation MobileApps and for instantiating quality models for either native MobileApps or web-oriented MobileApps, as we see later on.

Next, we examine several features and entities relevant for the quality design and evaluation of MobileApps with the ultimate goal to carry out evaluations and accomplish improvement in a MobileApp.

3. Featuring MobileApp Usability and UX

QinU, UX and Usability have all recently come to the research forefront due to a general shift in emphasis to satisfying the end user as part of the customer experience. For MobileApps (including native MobileApps and Mobile WebApps), they become even more important due to the significance of the user context. In particular the user’s activity at the time of usage, location, and time, amongst other influencing factors have significant impact on the quality of the user’s experience.

This section examines several elements relevant for designing and evaluating UX, Usability, Information Suitability, etc. for MobileApps.

These elements include sub-entities (e.g. widgets, menus, forms, etc.) and their associated attributes and
characteristics (e.g. Operability, Understandability, Functional Suitability, etc.). Some of the typical sub-entities for MobileApps that should to be considered for quality design and evaluation are:

- **Typing/input**: which includes search bars, and other data entry fields whereby the users should be assisted as much as possible to reduce errors and the ‘cost’ of typing. This includes such measurable attributes as default values, default value removal and shortcuts.

- **Entry widgets** such as carousels, drop down boxes and lists. System designers need to prevent the need for typing and reduce error rates by using widgets.

- **Sort, search and filter**: Special considerations are needed for MobileApps in order to reduce the workload and typing input. In addition, the small screen size makes it easy for the user to lose context, so attributes like typo tolerance and predictive contextual help are desirable.

- **Menus** should be limited, simple and easily navigated with a clear breadcrumb path showing where the user has come from and where they can go. This is mostly applicable to Mobile WebApps where a small screen limits the users’ context or field of vision in navigating from one place to another.

- **Forms and registration**: Forms need be clear with context sensitive help. The last thing you want is a user unable to complete a form because they didn’t quite understand one particular mandatory field. Either defaults, or help within the entry field giving an example of what goes in the field should be provided.

Regarding the aforementioned sub-entities of typical MobileApps, we can now consider some examples of sub-characteristics of Usability, Information and Functional Quality that are particular to MobileApps, for example:

i) **Learnability**: Through its various entities listed above such as menus and widgets, learnability can be designed into the MobileApp through defaults, facilitating predictive actions, context sensitive help, and so on;

ii) **Navigation**: The MobileApp’s ability to enable to a user to easily find the functionality or information that they need is critical. Not only do they need to easily find it, they need to do it fast. These are sometimes in alignment but not necessarily;

iii) **Operability**: This is a central sub-characteristic of usability. It means the degree to which a MobileApp has attributes that make it easy to operate and control. For example, controllability represents the degree to which users can initiate and control the direction and pace of the task until task completion. Easy to operate is related to those provided mechanisms which make entering data as easy and as accurate as possible while maintaining consistent usage and placement of controls even in different contexts and platforms of use;

iv) **Error handling**: Error prevention, error awareness, and error status are key attributes that need to be designed correctly to not only prevent errors, but enable the mobile user to recover with minimal effort. In addition, errors should be easily and quickly understood so that the user can move forward with their task and learn from the error;

v) **Understanding**: For MobileApps, because the screen is so small, it requires special consideration for understanding what the application is about, and what it does almost instantly. As mentioned previously, mobile users have a very short attention span so they must glance at the application and understand how it operates. For instance, an airline application, when they open it up, they already know they want to, either check a flight status or make a reservation, so the design must heavily consider this expectation so that there is reduced ramp up time;

vi) **Visibility**: Many factors determine whether or not the application is easily visible to a user. Depending on the context, different text colors and backgrounds can have a positive or negative impact. This sub-characteristic, while related to aesthetics is not identical. Remember that mobile users want to glance quickly and understand almost immediately and there may be glare on their screen if they are outdoors. This means appropriate usage and placement of text in appropriate format can impact the
user’s speed of comprehension greatly.

Table 2: Definition of Operability sub-characteristics (related to Usability) and potential attributes (in italic). Note that code numbers are only intended to show only hierarchical relationship dependences.

<table>
<thead>
<tr>
<th>Sub-characteristic / Attribute</th>
<th>2Q2U v2.0 Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Operability</td>
<td>Degree to which a product or system has attributes that make it easy to operate and control. Note: Same ISO definition (ISO/IEC 25010, 2011)</td>
</tr>
<tr>
<td>1.1 Controllability</td>
<td>Degree to which users can initiate and control the direction and pace of the task until task completion</td>
</tr>
<tr>
<td>1.1.1 Task workload reduction (synonym: Task brevity or conciseness)</td>
<td>Degree to which users can reduce the workload and cognitive effort by performing actions to complete a task in an abbreviated manner.</td>
</tr>
<tr>
<td>1.1.2 Task interruption support (synonym: Task operation cancel-ability)</td>
<td>Degree to which the task steps can be interrupted without harmful effects to the normal operation.</td>
</tr>
<tr>
<td>1.1.3 Undo action support</td>
<td>Degree to which the action can be undone without harmful effects to the normal operation. Note: This could also include timeouts where the user can return to their original location in the application.</td>
</tr>
<tr>
<td>1.2 Data Entry Ease</td>
<td>Degree to which mechanisms are provided which make entering data as easy and as accurate as possible</td>
</tr>
<tr>
<td>1.2.1 Defaults</td>
<td>Degree to which the application provides support for default information.</td>
</tr>
<tr>
<td>1.2.2 Mandatory entry</td>
<td>Degree to which the application provides support for mandatory data entry.</td>
</tr>
<tr>
<td>1.2.3 Control (widget) appropriateness</td>
<td>Degree to which the application provides the appropriate type of entry mechanism in order to reduce the effort required, i.e. reducing number of keystrokes and data entry through drop down list boxes and radio buttons which reduce typing needed.</td>
</tr>
<tr>
<td>1.3 Visibility (synonym: Optical Legibility)</td>
<td>Degree to which the application enables ease of operation through controls and content which can be seen and discerned by the user in order to take appropriate actions.</td>
</tr>
<tr>
<td>1.3.1 Control(widget)size appropriateness</td>
<td>Degree to which size of controls enable user to easily determine and understand their function as well as use the control.</td>
</tr>
<tr>
<td>1.3.2 Text size appropriateness</td>
<td>Degree to which text sizes and font types are appropriate to enable user to easily determine and understand their meaning.</td>
</tr>
<tr>
<td>1.3.3 Contrast visibility appropriateness</td>
<td>Degree to which text and controls colors provide contrast with background color so that they are easily visible. Note: This can be gauged considering background and foreground colors.</td>
</tr>
<tr>
<td>1.3.4 GUI widgets placement appropriateness</td>
<td>Degree to which the placement of widgets and text is appropriate and visible in any context.</td>
</tr>
<tr>
<td>1.4 Consistency</td>
<td>Degree to which users can operate the task controls and actions in a consistent and coherent way even in different contexts and platforms of use.</td>
</tr>
<tr>
<td>1.4.1 Control label consistency</td>
<td>Degree to which the labels of widgets/controls correspond to the actions they represent.</td>
</tr>
</tbody>
</table>
Table 2 shows for the Usability characteristic one of its sub-characteristic, i.e. the Operability sub-characteristic as prescribed in the ISO standard (ISO/IEC 25010, 2011) from the EQ standpoint. We have elaborated Operability, adding and defining new sub-characteristics and potential measurable attributes. Attributes can in turn be measured by means of metrics and evaluated and interpreted by means of indicators. In (Lew et. al. 2010) we have evaluated some Learnability and Operability attributes.

Finally, among other sub-characteristics and attributes to take into account for evaluating MobileApps is Information Conciseness, which in few words means shorter is better. In (Olsina et. al. 2012) Information Conciseness is an attribute of the Information Coverage sub-characteristic, which in turn is related to Information Suitability, and is defined as "degree to which the information coverage is compactly represented without being overwhelming". Note that some of these attributes are identified in the screens shown in the next section.

In alignment with the ISO 25010 'influences' relationship (see Figure 1), each of these system sub-characteristics/attributes can have an influence on the system in use for both the do-goals (Actual Usability characteristic and its sub-characteristics) and be-goals (Satisfaction characteristic and its sub-characteristics). Depending on the context, user, and task, the influence can impact the user's ability to operate, navigate and use the application efficiently (do-goals) or comfortably and with pleasure and utility perception (be-goals).

Thus, in designing a MobileApp, one must also take into account the decision to make a native MobileApp versus a Mobile WebApp. For instance, navigability, as described above, and congruence and consistency with the regular WebApp are significant. On the other hand, many widgets and other features of the mobile phone are not accessible to WebApps so they inherently start at a disadvantage regarding widget usage which, depending on the application, can impact system in use greatly.

4. Illustrating Quality Design and Evaluation for MobileApps

Given the aforementioned 2Q2U framework and mobile UX and usability concepts we can apply them to MobileApps. This section builds upon Section 3 and demonstrates the importance of a few of the factors through some examples whereby the context, system, and system in use come into play for MobileApps.

When designing MobileApp tasks for QinU, for example for evaluating Effectiveness and Efficiency characteristics, content and functions are embedded in the task design itself rather than as attributes of the application. And, as mentioned previously, UX do-goals for a MobileApp-in-use are heavily influenced by the workflow of the task. Because the most prevalent mobile tasks include deadline oriented, or time sensitive information, the workflow should be short in order to be effective and efficient to make up for the small screen size and user context (in a hurry with short attention span). User's are often executing tasks 'on the spot' and 'in the moment' so a delay, or a mistake is critical at that moment. Can you imagine getting stranded at an airport, and then looking up all the flights going out of an airport and not being able to search effectively because the search menu was designed only with one search parameter?

From this simple example, it is easy to see that QinU depends on many system sub-entities having particular sub-characteristics and attributes depending on the context, user and task. Note that when executing a particular task, there should be very few controls that could lead the user astray or toward a mistake (see for example the Fidelity Investment MobileApp screenshots in Figures 2.a and b). They purposely design the widgets and controls to be large, thumb friendly, so that the user can complete the task given the information needed as effectively and efficiently as possible. There is contextual help, or a simple drop down menu with limited choices in the fields that need to be filled in.

As mentioned for the typing/input sub-entity, defaults are extremely important to reduce the workload of the user. Defaults can be based on what the user has typed or submitted in the past (e.g., zip codes, names, addresses). Defaults are bound to be wrong sometimes and you don’t want to have a default that takes unnecessary effort to change. For example don’t use 0 as the default for a telephone number or zip codes,
code and don’t make users manually erase the text field character by character, by clicking the Delete key. Instead, erasing defaults with a single button that clears the entire field can save several seconds.

![Figure 2: a) Fidelity Investment MobileApp login screen; b) Fidelity Investments MobileApp find a ticker symbol task; and c)United Airlines MobileApp finding a flight](image)

In Figure 2.a, the MobileApp shows the default user name as well as the option to save it or not, giving the user a sense of controllability. Also, for typing/input and widget sub-entities, the United Airlines MobileApp shown in Figure 2.c, shows a carousel for data entry rather than typing. It also gives the user another option for input. Additionally, the screen is very clear for the task at hand. Notice that there is no other input or information not directly related to the task of Finding a flight.

Examining Figure 2.b, shows the Fidelity Investment MobileApp for the task of “Finding a stock ticker symbol”. Notice the context sensitive help in light grey color symbol or company name, which aids the user. Also, there is a list of past searches which helps the user as well. Both the ticker symbol and company name are shown to eliminate any confusion and prevent using the incorrect ticker symbol. The information is also presented in a clear and concise manner with no other information (information conciseness) to distract from the task at hand while features have been designed to reduce errors and increase efficiencies and task completion rates. The last thing Fidelity wants is for a user to not complete this task which may affect executing a trade.
Figures 3.a and b show the Embark application for the Washington DC Metro train system. Notice that (Figure 3.a) for the task of “finding a station”, the user is given context sensitive help below the search bar. There is also light grey letters in the search bar to make sure the user is not lost, reminding the user what they are trying to do. Also notice that the search bar is quite long, thus providing the user with ample space to type in their desired location. On the other hand, for the same entity, you can see from Figure 3.b that the errors are not comprehensible thus very difficult for the user to recover and move forward with their task. Lastly, Figure 3.c shows a search function implemented in the Apple App Store. For the App Store, a user in search of ‘tvguide’ accidently typed ‘tvgiude’ and got no results. Thus in terms of typo tolerance and error prevention, it would rate poorly for that particular task.

These examples demonstrate MobileApp sub-entities such as forms, menus and search boxes and why design principles, i.e. mobile system factors can significantly influence the UX and other system in use factors. With a screen so small, it is easy to see why concise information, contextual help, etc. are critical attributes.

Note that a systematic and specific instantiation of quality models for MobileApps -as we did in (Lew, Olsina et. al. 2012) for the JIRA WebApp’s QinU and EQ- will be documented in a separate manuscript.

In (Lew, Olsina et. al. 2012) we elaborated SIQinU (Strategy for understanding and Improving Quality in Use), which is a specific-purpose and context-sensitive strategy to incrementally and continuously improve a WebApp-in-use’s QinU by means of mapping actual usage problems to measurable EQ attributes –that are inherent to a WebApp-, and then by performing improvement actions that enable evaluators assessing the gain both at EQ and QinU levels. SIQinU is a specific-purpose strategy because it can be used to evaluate the quality for only system-in-use and system entity categories by using their instantiated QinU and EQ models respectively.

Basically, the strategy collects user task data from log files (that were derived through for example adding snippets of code in a real WebApp-in-use that allow recording user activity), with the aim to derive attributes, metrics and indicators for QinU, thus leading us to understand the current QinU satisfaction level met. Then, by performing a preliminary analysis, EQ requirements that can affect QinU are derived, followed by proposed recommendations for improvement. After performing the changes on the WebApp, and after conducting studies with the same user group in the same daily environment (context) with the new (and improved) system version, an assessment of the improvement gain can be gauged.
5. Conclusions and Future Work

In summary, we have characterized relevant features of MobileApps with regard to usability and UX analyzed them using our previously proposed 2Q2U v2.0 quality models. In the process, we looked at MobileApp usage context and its impact in MobileApp design when considering quality characteristics and attributes for system and system-in-use quality evaluation. Lastly, we illustrated some example MobileApps to show their design features could have a significant influence on the system quality and the user's perception of quality, i.e. system in use and UX.

Our proposal shows the usage of 2Q2U v2.0 as an integrated approach for modeling requirements for external quality and quality in use (i.e., actual usability, satisfaction and user experience) for MobileApps.

Another manuscript will expand the work further including the definition of attributes, metrics, and indicators for user group types performing specific tasks in real MobileApp context of use. Ongoing research is focused on further utilizing the 2Q2U v2.0 framework for systematic instantiation of quality models for MobileApps –as we did in (Lew, Olsina et. al. 2012)- in order to provide foundations when modeling and understanding the relationships among EQ, QinU, actual usability and UX. In doing so, our end goal is improvement. That is, to be able to directly use the system-in-use quality evaluation to directly improve the design of the MobileApp system. This concern has often been neglected in the literature, but may help improve quality design recommendations and ultimately to increase MobileApp UX as a whole.
References


