A Cross-Platform Usability Measurement Model
Khalid Majrashi and Margaret Hamilton

Abstract—Ubiquitous technologies are changing our lives. We are becoming more connected and able to conduct our computing tasks anywhere at anytime from any device. Vertical interaction with an individual interactive system is no longer the only way to achieve tasks. Currently, users can interact horizontally with multiple user interfaces to achieve their tasks. This has created a need for measuring usability of multiple interactive systems, concerning users horizontal interaction beside their vertical interaction. In this paper, we surveyed the actual meanings and interpretations of usability and its attributes across several standards and models. We found that the existing usability standards and models do not consider horizontal usability aspects. Therefore, taking into consideration the characteristics of user interaction with multiple interactive systems, a hierarchical model, which is called Cross-Platform Usability Measurement (CPUM), has been developed. This model decomposes cross-platform usability into 12 factors. These factors were further decomposed into measurable criteria, and finally into specific metrics.

Index Terms—Cross-platform services, horizontal usability, software quality, user experience.

I. INTRODUCTION

The emergence of several computer devices including mobile phones, personal digital assistants (PDAs) and pocket PCs has resulted in a new computing evolution [1]. A large number of existing devices are increasingly being introduced as an alternative to old style computers, such as smartphones and tablets. Current handhelds can provide high resolution screens, monitors with touch screen capability and overall performance that are adequate for some users to stop using PCs. Handhelds allow accessing business databases, editing advanced accounting functionalities, preparing PowerPoint slides, as well as having the advantage of accessing the Web. The cloud-computing paradigm allows users to access services (e.g., software as a service) from a range of different devices, enabling users to migrate their interactions cross-device. The long-term storage issue with handheld devices can now be solved with cloud storage (e.g., for documents: Google storage (drive.google.com) and for photos: iCloud (www.icloud.com)). This allows users to use handhelds with more functions that rivaling PCs. Furthermore, private cloud services (e.g., www.tomido.com) allow accessing all files on PC’s via handhelds web browsers or mobile applications, which means decreasing the reliance on PCs. Furthermore, Desktop as a Service (DaaS) in which the back-end of a virtual desktop infrastructure (VDI) is hosted by a cloud service provider allows on-demand and high-performance access to virtual desktops, including Windows OSs, the Web and SaaS apps or IT services from any device, whether a smartphone, tablet, notebook or PC. Therefore, the dominance of PCs is not the same as it was in previous decades, because mobile devices are currently a common choice when it comes to accessing a variety of applications and services [1].

With the new capabilities of handhelds, and the new computing paradigm (e.g., cloud computing), users can achieve their tasks or goals horizontally (using different devices or platforms). For example, for sending an email, a user first starts using a PC app, and then switches to the mobile phone app to complete the task. However, the variation in devices that can be used to achieve horizontal goals makes a huge barrier to pervasive computing where users are drowning in the diversity [2]. The paradigm of measuring how easy a user achieves his or her goals cross-platform is still unidentified in the literature. The current usability measurement models are focused more on measuring how easy a single user interface can be used by intended users. This means that in the planning phase, we recruit samples of intended users of the user interface, and design tasks to be achieved using the user interface within an individual platform. In testing phase, we observe those samples of intended users when interacting with the user interface, and performing the tasks we provide them. Therefore, we can find usability issues, and draw a conclusion about the user interface in terms of its efficiency, effectiveness, as well as user’s perception of it.

There is a difference among vertical and horizontal usability terms. The term vertical usability deals with the traditional usability aspects of an individual system within a specific platform. This type of usability can be achieved by applying specific design guidelines issued by system manufacturers (i.e., OS X Human Interface Guidelines 1) to assist designers in developing applications within specific platform [3]. Designers can also rely on specific usability principles, which can guide them in making usable user interfaces (for example, Nielsen’s ten usability heuristics [4]).

Developing the same service (a user interface) for multiple platforms (e.g., Android, iOS, Window, OS X) introduced new requirements for usability cross-platform (e.g., consistency of the service cross-platform). Seffah, et al. [3] referred to this dimension of usability as horizontal usability. The same concept has been used with different naming

conventions, for example, Denis and Karsenty [5] referred to it as *inter-usability* to designate the ease of use of multiple user interfaces whereby users can reuse their knowledge and skills for a given functionality when switching to other devices. The term *inter-usability* has been used in line with the original notion of inter-operability, which is defined as the possible uses of software programs on a number of computer platforms. *Multi-Device Usability* also refers to usability cross-platform [6].

The usability of services cross-platform is a new research theme within the software engineering and HCI communities. According to our knowledge, there is no cross-platform usability measurement model that can be used to address usability aspects in a methodical and structured way. There is a need for considering cross-platform (horizontal) usability aspects in parallel with traditional (vertical) usability when designing systems that are expected to be used vertically (achieving a goal using an individual interactive system) or horizontally (achieving a goal cross multiple interactive systems). In response to this need, we develop a usability measurement model for cross-platform services in this paper. Our model is designed to enable the evaluation of the capabilities of cross-platform services to support user horizontal interaction. The cross-platform usability measurement model can be a part of service engineering, which may take over the term software engineering in the near future.

For producing our model, we review different usability definitions, software quality and usability attribute models, and cross-platform services characteristics. We then develop our model, which concentrates on cross-platform contexts of use, cross-platform usability attributes, measurable criteria and metrics.

This paper is organized as follows. In Section II, we present a review of usability definitions, standards and models. Section III highlights terms that are associated with multiple interactive systems and illustrates different configurations of devices concerning the level of data and functions available to users. The CPUM model is presented in Section IV. In Section V, an update to current usability-testing procedures to suit cross-platform usability measurements is presented.

II. REVIEW OF USABILITY DEFINITIONS, STANDARDS AND MODELS

In this section, we present different definitions of usability and identify several usability attributes from various different standards, and models.

A. Usability Definitions

Usability is an important attribute of software quality that has several methods and metrics that can be used to assess how easy a user interface is to use. One goal of usability is to bridge the gap between human goals and technology [7]. For bridging the gap, the needs and their behaviors of humans should be understood to make interactive systems conform to what they want and expect.

Although usability has its academic origins in human computer interaction and software engineering communities, the term is still undefined in a consistent way. Bevan [8] outlined that different people and different standards interpret the term usability differentially. A few common definitions of usability are listed below:

- **Shackel’s definition:** “[a system’s] capability in human functional terms to be used easily and effectively by the specified range of users, given specified training and support, to fulfill a specified range of tasks, within the specified range of environmental scenarios” [9].
- **IEEE definition:** “The ease with which a user can learn to operate, prepare inputs for, and interpret outputs of a system or component” [10].
- **ISO/IEC 9126 definition:** “A set of attributes that bear on the effort needed for use and on the individual assessment of such use, by a stated or implied set of users” [11].
- **Preece’s definition:** “a measure of the ease with which a system can be learned or used, its safety, effectiveness and efficiency, and the attitude of its users towards it” [7].
- **ISO 9241-11 definition:** “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” [12].

The definitions of usability have been focused on vertical usability, which can be measured based on user interaction with a specific system within a specific platform. However, our concern is about usability of services cross-platform, where these services can be used in combination in order to achieve specified goals in horizontal manner from different contexts of use (e.g., moving, standing and seating). In comparison with vertical usability, horizontal usability can be measured based on user interaction with multiple interactive systems when using different platforms or devices.

The different interpretations of usability as a term across academic and industry societies have impacted on the ability to identify standardized usability attributes in a consistent way over time. Therefore, in order to define usability of cross-platform services precisely with considering all possible usability aspects, we need to review usability attributes from different models and standards, and identify characteristics of multiple interactive systems.

B. Usability in Software Quality Models

Usability has been introduced in different software quality models. In 1977, McCall’s Quality Model (known as the General Electrics Model of 1977) has been one of the most recognized quality models in the software engineering discipline. McCall seeks to address the gap between users and developers by paying attention to a range of software quality characteristics (called factors in the model) that reflect both the users’ views and the developers predilections [13]. The structure of the McCall’s quality model consists of three major layers: Factors, Criteria and Metrics (FCM). Usability is described in McCall’s quality model as a major quality factor, which comprises three criteria: operability, training and communicativeness. These are defined as follows:

- **Operability** is the ease of operation of the given system
- **Training** is to what degree the software supports in enabling new users to apply the system
- **Communicativeness** is to what degree software is designed in accordance with the psychological
characteristics of users

Metrics in McCall’s quality model are associated with the criteria. The following are examples of metrics associated with the usability criteria:

- Time needed for a user to carry out operation(s) of the software.
- Time for a new user to obtain a basic understanding of elements of the software
- Number of negative comments from new users concerning ergonomics and human factors.

The use of the metrics in McCall’s model allows concrete measures of usability. However, there are important usability attributes that were not included explicitly in the model such as memorability and safety.

In 1978, Boehm et al. [14] created a quality model to quantitatively estimate the quality of software. A predefined set of quality attributes and metrics have formed the structure of the model. It is comprised of three top-level characteristics, seven intermediate-level characteristics and 15 lowest-level characteristics that actually lead the way to the overall quality level. The model suggested treating the quality of a software product via three perspectives: 1) Product utility to care for how well, easily, reliably and efficiently end-users can use the software product as-is. It covers reliability, efficiency and human engineering criteria. 2) Portability to address if a user can continue to use the software product as soon as the environment has been changed 3) Maintainability to address how easy it is to understand, modify and retest the software product. It covers testability, understandability as well as modifiability criteria. In comparison to McCall’s model, Boehm’s model did not describe usability and its metrics clearly. Furthermore, some metrics (e.g., communicativeness) can be associated with different quality factors. This creates a difficulty in understanding how ease of use should be measured in Boehm’s model.

In 1992, FURPS model (proposed by Grady [15]) considered five characteristics that outline its name: Functionality, Usability, Reliability, Performance and Supportability. The model separates characteristics into two different categories of requirements: functional and non-functional requirements. Usability came under non-functional requirements. The model suggested different factors to be included under usability characteristics, encompassing aesthetics, consistency, human factors, online and context sensitive help, wizards, user documentation and training materials. FURPS model did not include metrics to measure usability factors.

In 1995, Dromey’s model [16] suggested a working framework for assessing different phases in the software engineering life cycle including requirement, design and implementation. One of the advantages of this model is the definition of an explicit process for building quality-carrying properties into software, which in turn involves specific quality attributes. The product properties for the implementation quality model include: correctness, internal, contextual and descriptive. Usability came as property attributes under descriptive which indicates that it is a product attribute. The model did not explain how to measure usability and its definition was too abstract.

C. Usability in ISO and IEEE Standards or Models


ISO 9241-11 (1998) [12] described three major usability measurement attributes: effectiveness, efficiency and satisfaction, which can help in determining whether a product can be used by specific users to achieve specific goals. The standard described effectiveness as the accuracy and completeness of achieving a specific goal; efficiency as the resources that have been expanded in relation to accuracy and completeness of tasks; and satisfaction as the comfort and acceptability of use. Although these three attributes are largely recognized in the literature, as well as in the common industry format of usability testing [21], there are some limitations that have been identified regarding these measurement attributes. For example, it is not a clear how these three measures correlate with each other [22], and so we cannot determine, for example, whether the completeness rate of tasks plays any role in increasing or decreasing the user satisfaction rate.

Quality of product has been revised in the series of ISO/IEC 9126 [17]-[20]. The new standard distinguished between three different approaches to product quality: internal quality concerns non-executable parts of software which can be measured based on the source code (e.g., path length); external quality concerns properties of the code when executed, and can be measured based on execution-relevant metrics (e.g., response time); and quality in use which concerns the needs of end users, and can be measured according to whether a software meets user needs (such as effectiveness). Bevan [23] described the relationship between the three quality approaches in terms of cause and effect. For example, the cause of having software with a good quality in use should be that the external attributes are at an excellence level, and this excellence level can only be achieved via a good level of internal attributes.

The ISO/IEC 9126-1 (2001) [17] standard divided usability into five factors: understandability, learnability, operability, attractiveness and usability compliance. In ISO/IEC 9126-4 (2004) [20], a new two-layer model (quality in use model) has been described which is composed of characteristics and quality measures. It described quality in use as the user’s view of software quality when the product is used in a certain environment and in a specific context. The quality in use model is decomposed into high-level quality factors: effectiveness, productivity, safety, and how it satisfies users.

In 1990, the Institute of Electrical and Electronics Engineers (IEEE) published the standard glossary of software engineering terminology and included comprehensibility, ease of learning and communicativeness as the major usability attributes [10].

D. Usability in Other Standards or Models

In 1991, Shackel [24] was one of the first researchers who proposed a list of measurement attributes for usability. He stated that usability can get measured using the operational
criteria of effectiveness, learnability, flexibility and attitude. The following Table I shows examples of metrics associated with the usability criteria in the Shackel’s model.

<table>
<thead>
<tr>
<th>Usability Criteria</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effectiveness</td>
<td>Concerning speeds and errors, the necessary range of tasks needs to be accomplished at a level that is better than some required level of performance.</td>
</tr>
<tr>
<td>Learnability</td>
<td>A user should be able to use the system after a specific time of first self-usage or proper training.</td>
</tr>
<tr>
<td>Flexibility</td>
<td>With flexibility enabling adaptation to a certain selected percentage variation in tasks and or environments beyond those at the beginning specified.</td>
</tr>
<tr>
<td>Attitude</td>
<td>With an acceptable level of tiredness, discomfort, frustration and personal effort.</td>
</tr>
</tbody>
</table>

Bevana et al. [25] advised using three characteristics for accurate measurements of usability: user performance, satisfaction and acceptability. They emphasized that any kind of modification within the characteristics of the product, system, user, task or environment might possibly generate a variance in usability measurement.

Shneiderman [26] and Nielsen [27] used the same attributes for measuring usability, which are learnability, efficiency, memorability, errors, and satisfaction. These two models used memorability as the major usability factor instead of learnability as in other models (e.g., in the model proposed by Shackel [24]). They referred to it as the level in which the system can support users to remember how to use its functions and services after a period of time of not using it.

Beside the usability attributes that have been already discussed, Preece et al. [7] reflected more on safety as an important usability attribute due to the fact that interactive systems can not only be found in the stationary context of use with desktop machines. They can also be used in other contexts, for example, in the medical arena and air-plans, and this introduces the need to allow users to use the system safely without affecting their life or other people’s life or even preventing users from making undeliberate damage to their resources in the system.

It is evident so far that there were two main goals in the reviewed measurement models. Firstly, the usability goal, which can be measured, for example, using attributes such as efficiency and effectiveness. Secondly, the user experience goal which can be measured with satisfaction. Rogers et al. [28] proposed more attributes to be used to measure the user experience goal besides satisfaction; for example, the system should be enjoyable, fun and motivating.

A summary of various reviewed usability attributes over four decades are given in Table II, which are classified based on publication year.

The reviewed usability attributes concern the usability of an individual interactive system. For example, the capability of software to facilitate the user in accomplishing the task with precision and completeness. However, they do not take into consideration the usability of multiple interactive systems when a user interacts with them to achieve a horizontal goal.

<table>
<thead>
<tr>
<th>Year</th>
<th>Source</th>
<th>Usability Attribute(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1977</td>
<td>McCall et al. [13]</td>
<td>Operability, training, communicativeness</td>
</tr>
<tr>
<td>1985</td>
<td>Butler [29]</td>
<td>User performance</td>
</tr>
<tr>
<td>1986</td>
<td>Mills et al. [30]</td>
<td>Speed and accuracy of task execution</td>
</tr>
<tr>
<td>1988</td>
<td>Gould [31]</td>
<td>System performance (e.g., reliability), system functions, user interface, Outreach Program (e.g., on-line help), Modifiability, Language Translation,</td>
</tr>
<tr>
<td>1994</td>
<td>Nielsen [27]</td>
<td>Learnability, memorability, errors, satisfaction</td>
</tr>
<tr>
<td>2001</td>
<td>Donyaee and Seffah [33]</td>
<td>Effectiveness, efficiency, satisfaction, productivity, safety, internationality, accessibility.</td>
</tr>
<tr>
<td>2002</td>
<td>Bring, et al. [34]</td>
<td>Functionally correct, efficient to use, easy to learn, easy to remember, error tolerant, subjectively pleasing.</td>
</tr>
<tr>
<td>2005</td>
<td>Shneiderman and Plaisant [36]</td>
<td>Time to learn, speed of performance, rate of errors by users, retention over time, subjective satisfaction.</td>
</tr>
</tbody>
</table>

### III. Multiple Interactive Systems and Multi-Device Data, Services, and Functions

A brief review of multiple interactive systems and multi-device data, services, and functions is presented in this section.

#### A. Multiple Interactive Systems

In HCI, several terms have been used to describe interactive systems encompassing more than one device, or platform that are used to deliver services (user interfaces) to end-users. Seffah and Javahery [37] used Multiple User Interfaces (MUI) to describe systems where users can access the same information and services from different computing platforms (e.g., hardware [i.e., office desktops, and mobile phones] and OS’s [i.e., windows, and iOS]). Distributed User Interface (DUI) is widely used by HCI researchers for referring to the same concept (see [38]-[41], and [42]). Along similar lines, some researchers (e.g., Olsen et al. [2]) used the Cross-Modal term when they referred to user interaction cross-platform.

#### B. Multi-Device Data, Services and Functions

Technical constraints of mobile devices prevent accessing the full advantages of a large amount of data, services and complex functions. Denis and Karsent [5] outlined three degrees of device redundancy, which demonstrate levels of presenting data and functions cross-device. The first configuration is redundant, where all the interactive systems cross-device can allow access to the same.
data and functions. In this configuration, a new user may tend not to fully comprehend this redundancy as a result of the unique visual appearance or arrangement of the service via each device.

The second configuration is exclusive where each interactive system in each device gives access to different data and functions. It is an issue if a user at this level believes that different data and functions can be accessed via different devices. In addition, it is known that previous experience can affect the system on hand [43], and therefore it is possible that previous experiences with services that implemented the first configuration can affect the usability of current services that apply the second configuration. This is because users can be frustrated if they are not able to access certain data or perform specific functions as they were expecting based on their previous experiences.

The most used configuration is complementary, whereby the interactive systems in all devices have a zone of shared data and functions, but one or more of the devices offer the ability to access data or functions that are inaccessible on the other device(s). Besides the issues that users might encounter with the redundant and exclusive configurations, users may become more confused when using complementary configurations. If they found a certain data or function which is available cross-device from the shared zone, users may build a representation model of the services that all data and functions are available as well. This may frustrate the user when discovering unavailable data and functions.

Increasing the number of services that the user interacts with cross-device may lead to the user becoming more confused and frustrated. This is because the more services that users interact with, the greater the possibility of having different configurations, which can affect user understandability of systems cross-device. The following Fig. 1 illustrates some possibilities of device configuration at 3-level services.

![Fig. 1. Possibilities of degree of redundancy at 3-level services cross-device.](image)

With cloud computing, it is now possible to overcome obstacles in mobile computing. For example, the cloud can help reduce the running cost for compute-intensive applications that require a long time and large amounts of energy when performing on the limited-resource devices such mobile phones [44]. As a consequence, it is possible to have full access to data functions and services through different mobile or non-mobile devices.

IV. CROSS-PLATFORM USABILITY MEASUREMENT MODEL

Reviewing usability attributes from different standards and models, and identifying characteristics of multiple interactive systems allowed us to define cross-platform usability as:

*The extent to which services cross-platform can be used by specified users to achieve specified horizontal goals from different contexts of use with acceptable level of several measurable factors including efficiency, effectiveness, learnability, memorability, productivity, accessibility, understandability, satisfaction, universality, helpfulness, safety, and visibility.*

This definition guided us through the process of developing cross-platform usability measurement model. We amend the current usability measurement models including contexts of use factors, usability attributes (factors), measurable criteria and metrics. Just like other software engineering models (e.g., [13], [14]), we use a hierarchical structure for our model starting from top level factors (e.g., learnability), going through measurable criteria (e.g., task completeness), and reaching to the metrics (e.g., after certain time of usage, the percentage of users who are able to complete a certain task cross-platform). According to our knowledge, CPUM is the first model concerning horizontal usability aspects. It will serve as guidance for UX evaluators for testing usability cross-platform with users.

The structure of this section is as follows. We start by describing possible factors of contexts of use that may affect usability measurement cross-platform, and then present usability factors, measurable criteria and metrics in CPUM’s model.

A. Contexts of Use

The determination of the usability level of a certain product or service is not only based on measuring usability attributes of the product or service itself. We should be aware of other factors that can affect usability such as the context in which it is used, including the particular user’s tasks and environments (e.g., physical environment) [8], [12].

Different user interfaces in different devices are designed to be used in varying contexts. As we have discussed earlier, the user goal can be achieved using different user interfaces via different devices, which means that the goal also can be achieved in different contexts of use. However, it is also possible to achieve the goal using different user interfaces via different devices in a specific context of use. Öquist, et al. [45] identified different contexts of use for multiple user interfaces. Those contexts of uses are the stationary context of use (exemplified by the desktop computer), seated context of use (exemplified by the laptop users), standing context of use (devices can be hold with one hand [e.g., palmtop]) and moving context of use (associate with smaller handled devices).

In measuring cross-platform usability, characteristics need to be considered for each interaction session (interaction session means interacting with a service from an individual platform as a part of other interactions with the service cross-platform for achieving a single goal). The context of use developed in [12] has been modified in this paper to suit measuring usability cross-platform. We have arranged context of use into three main factors: user and task, environmental, and situational factors (see Fig. 2). The
characteristics of different factors need to be considered for different interaction sessions.

### TABLE III: CONTEXT OF USE FACTORS AND CHARACTERISTICS

<table>
<thead>
<tr>
<th>Psychological</th>
<th>Motivation, cognitive style, and attitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>General: speed of learning, computer literacy, reading level.</td>
</tr>
<tr>
<td>Knowledge and experience</td>
<td>Specific: platforms experience (e.g., experience with iOS OS), devices experience (e.g., experience with touch screen in mobile device), task experience (task experience with any of tested services), typing skills, service experience (participant has already used the service within specific platform), domain experience (participant has already used a similar service in the same domain of tested services)</td>
</tr>
<tr>
<td>Physical</td>
<td>Task structure, task clarity, indication of which part of task needs to be achieved with each service, tasks in realistic sequence with tested services cross-platform, tasks in a reasonable length, tasks in the user words, tasks have no technical terms, and short.</td>
</tr>
<tr>
<td>Task Complexity</td>
<td>Training level: Refers to the level a user may receive with any of services cross-platform.</td>
</tr>
<tr>
<td></td>
<td>External tools: The use of, for example, paper and pen when carrying out a horizontal task (horizontal task encompass sub-tasks to be achieved cross-platform)</td>
</tr>
<tr>
<td></td>
<td>Frequency: Frequency of conducting a horizontal task (cross-platform), or vertical task (within individual platform).</td>
</tr>
<tr>
<td></td>
<td>Time: Refers to overall horizontal task time, needed time to switch between devices, each sub-task time within individual platforms.</td>
</tr>
<tr>
<td>Technical</td>
<td>Speed of each interactive system cross-platform, speed of device, memory capacity, output and input devices, network connections, speed of synchronisation, operating system and compatibility issues.</td>
</tr>
<tr>
<td>Physical</td>
<td>Noise level, privacy, safety, temperature, and light</td>
</tr>
<tr>
<td>Organizational</td>
<td>The organisation’s culture and policies.</td>
</tr>
<tr>
<td>Social</td>
<td>The level of assistance a user receives, and interruption (local interruption [e.g., face to face communication] or remote interruption [e.g., talking via a mobile phone]).</td>
</tr>
<tr>
<td>Situation</td>
<td>Refers to whether a user uses a service(s) in seating, moving, driving, or standing situations.</td>
</tr>
</tbody>
</table>

Each factor in the contexts of use has different characteristics, which need to be considered in terms of their effects on usability measurements cross-platform. In the following Table III, some characteristics that have been defined in [12] are amended and reconsidered for cross-platform usability measurements:

#### B. Cross-Platform Usability: Factors, Measurable Criteria and Metrics

Based on our review of the usability attributes from the literature (described in section II), we selected 12 usability factors to be part of CPUM. The selection was based on whether or not a usability attribute can be redefined and reconsidered for cross-platform usability.

Fig. 3 shows the adopted and redefined usability factors for cross-platform services. The factors are explained in the following with measurable criteria and metrics:

1) Cross-Platform Efficiency is the capability of cross-platform services to enable users to achieve tasks using any number of services within a specified time in a specified context of use. Traditionally, efficiency as a usability attribute concerns an individual user interface in terms of how it supports achieving vertical tasks efficiently. This is not applicable to measuring efficiency of services cross-platform when users attempt their tasks horizontally. This is because measuring the efficiency of each service separately and adding the measurements of each service together may not help in determining whether a user can achieve a horizontal task efficiently. Therefore, a user must attempt a horizontal task to enable us to determine how efficient the services cross-platform are, to support cross-platform interaction. To measure cross-platform efficiency, we propose the following measurable criteria, which are associated with the metrics.
• Time to execute a task: it is the spent time in executing a horizontal task. The following equation (Eq. (1)), which is described in Table IV, can be used to measure the time taken to execute a task horizontally:

\[
T_{\text{execution}} = \sum_{i=1}^{n} T_{\text{exec}_i} = \left( \sum_{i=1}^{n} T_{\text{entr}_i} + \sum_{i=1}^{n} T_{\text{exit}_i} + \sum_{i=1}^{n} T_{\text{sw}_i} + \sum_{i=1}^{n} T_{\text{ex}_i} \right)
\] (1)

<table>
<thead>
<tr>
<th>TABLE IV: DESCRIPTION OF EQUATION (1), THE EXECUTION TIME FOR A HORIZONTAL TASK</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \sum_{i=1}^{n} T_{\text{ent}_i} )</td>
</tr>
<tr>
<td>( \sum_{i=1}^{n} T_{\text{exit}_i} )</td>
</tr>
<tr>
<td>( \sum_{i=1}^{n} T_{\text{sw}_i} )</td>
</tr>
<tr>
<td>( \sum_{i=1}^{n} T_{\text{ex}_i} )</td>
</tr>
<tr>
<td>( \sum_{i=1}^{n} T_{\text{syn}_i} )</td>
</tr>
</tbody>
</table>

- Minimal number of actions: the capability of cross-platform services to help users achieve their tasks using any number of services in a minimum number of steps. The following equation (2), which is described in Table V, can be used to measure the minimal number of actions cross-platform:

\[
M_{\text{action}} = \sum_{i=1}^{n} S_{\text{total}} - \sum_{i=1}^{m} UnS_{\text{total}}
\] (2)

<table>
<thead>
<tr>
<th>TABLE V: DESCRIPTION OF THE EQUATION (2) FOR THE MINIMAL NUMBER OF ACTIONS FOR HORIZONTAL TASK</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \sum_{i=1}^{n} S_{\text{total}} )</td>
</tr>
<tr>
<td>( \sum_{i=1}^{m} \text{Un}S_{\text{total}} )</td>
</tr>
</tbody>
</table>

- Navigation: the degree to which cross-platform services can allow users to move around easily, as well as provide a consistent way of navigation cross-platform, so that the user does not get confused when transferring their tasks between devices. The following metrics are proposed to measure navigation cross-platform:
  a) After switching from one device to another, how much time a user spends to find or navigate to same page or view, which the user was exploring before the transition.
  b) After switching from one device to another, the number of times a user expresses that he or she feels lost in the service when attempting a horizontal task.
  c) After switching from one device to another, the number of times a user navigates to irrelevant elements in the horizontal task.

- Help: the degree to which users can use features, data and functions of cross-platform services without the need for assistance from others or using services documentation. The following metrics are examples of measures that can be used to determine if a user needs help when attempting a task cross-platform:
  a) After switching from one device to another, how many times a user asks for help or browse user documentation for the purpose of continuing the task that the user was trying to accomplish within the previous interacted-with user interface.
  b) The percentage of help that users receive for accomplishing each task cross-platform.

2) Cross-Platform Effectiveness is the capability of cross-platform services to enable users to achieve tasks with accuracy (low level of errors) and completeness (successfully achieving tasks). User interface in each device should be designed with consideration to task transition between devices, so that when a user migrates from a specific device to another, he or she can accomplish the task effectively. The following measurable criteria can be used for the purpose of measuring cross-platform effectiveness:
  a) Error rate: concerning horizontal interaction with multiple user interfaces, we define error rate as the count of incorrect actions made by users when they are trying to accomplish a cross-platform goal.
  b) Task completion: is the degree to which cross-platform services allow users to perform or complete tasks and achieve their goals. The completion rate for cross-platform usability can be measured using the following:
    a) Completed successfully: the task is completed successfully cross-platform without errors.
    b) Completed successfully with errors: the task is completed successfully cross-platform with some errors.
    c) Completed partially: only sub(s) of the task is completed using \( n \) number of devices.
    d) Completed wrongly: a user thought it was complete but I wasn’t.
    e) Given up: user gives up on task.
    f) Stopped task: the task is stopped by the moderator because of exceeding time.

Based on the above guidance for task completion, we developed other measurable criteria:
  a) Accuracy of tasks completion cross-platform which refers to the percentage of the total number of sub-tasks that are completed successfully with errors from all tested services divided by the total number of sub-tasks that are completed successfully without errors from all tested services (see (3)).

\[
T_{\text{accuracy}} = \frac{\sum_{i=1}^{n} T_{\text{subtasks with errors in service}_i}}{\sum_{i=1}^{n} T_{\text{subtasks with no errors in service}_i}} \times 100
\] (3)

\[
T_{\text{effectiveness}} = \frac{W \left( \sum_{i=1}^{n} T_{\text{subtasks in service}_i} \right)}{S \left( \sum_{i=1}^{n} T_{\text{subtasks in service}_i} \right)} \times 100
\] (4)

- Tasks effectiveness cross-platform can also be measured based on the percentage of the total number of
sub-tasks that are completed wrongly from all tested services divided by the total number of sub-tasks that are completed successfully in each service (see (4)):

3) Cross-platform Learnability: the extent to which services can be learned efficiently. The following measurable criteria and their associate metrics can be used to measure cross-platform learnability.
   - Task completeness and training:
     a) Percentage of users who completes horizontal tasks after receiving training with only one user interface within specified device.
     b) Percentage of users who completed horizontal task after receiving training with all user interfaces cross-device.
     c) Percentage of users who completed horizontal task without receiving training.
   - Task completeness after a certain time of usage:
     a) Percentage of users who complete horizontal task efficiently after a certain time of interacting with all services cross-device.
     b) Percentage of users who complete horizontal tasks with a low rate of errors after a certain time of interacting with all services cross-device.
     c) Number of users who review documentation and help resources to be able to continue their horizontal task after transferring to use the service from another device.
   - Task completeness after a certain time of usage:
     a) Percentage of users who complete horizontal task efficiently after a certain time of interacting with all services cross-device.
     b) Percentage of users who complete horizontal tasks with a low rate of errors after a certain time of interacting with all services cross-device.
     c) Number of users who complete horizontal tasks efficiently after a certain time of interacting with only a service within a specified device.
     d) Over a certain time of usage, the percentage of horizontal tasks which are completed by users without assistance in comparison to initial usage.
   - Learnability and user performance:
     a) Comparing users performance cross-platform for novice and expert users.
     b) Comparing novice users performance cross-platform in two phases: initial usage and after a certain time of use.

4) Cross-platform Memorability: we define memorability attributes of cross-platform services as the level to which user interfaces cross-platform help users to remember elements and functionality when interacting with a specific service within a specific platform based on their previous usage of the same service within another platform. The following measurable criterion can be used to measure cross-platform memorability:
   - Minimal memory load: the capability of cross-platform services to support users so that they are not required to keep minimal amounts of information in mind when transferring from one device to another in order to achieve a specified horizontal task. The metrics below can be used to determine the minimal memory load:
     a) Number of users who express that they feel the need to have some amount of information in mind to be able to continue their activities after transition.
     b) Percentage of users who stop completing a task because they are unable to remember a certain thing after migrating their tasks to another device.

5) Cross-Platform Productivity concerns user productivity when attempting goals using services cross-platform. Period of productivity is a possible measurable criterion for the productivity factor. We propose the following (5), which is described in Table VI, to measure the period of productivity:

\[
P_{\text{productivity}} = \frac{\sum_{i=1}^{n} T_{\text{productive}} - \sum_{i=1}^{n} T_{\text{unproductive}}}{\sum_{i=1}^{n} T_{\text{productive}}} \times 100 \tag{5}\]

| \(\sum_{i=1}^{n} T_{\text{productive}}\) | Productive time: is the proportion of the time a user spends on sub-tasks progressing towards the task goal cross-device, regardless of whether the task goal is eventually achieved or not.
| \(\sum_{i=1}^{n} T_{\text{unproductive}}\) | Unproductive time is the proportion of the time a user spends on seeking or reviewing help or trying to recover from errors in relation to sub-tasks when progressing towards the task goal cross-device, regardless of whether the task is eventually achieved or not.

6) Cross-Platform Accessibility: it is the degree to which people with any type of disability can use the services cross-platform to achieve their horizontal goals. The level needed to implement the accessibility principles in each user interface, and the consistency of the accessibility settings cross-platform, need to be considered when designing services cross-device. The following are examples of measurable criteria with metrics for cross-platform accessibility.
   - Task completeness and accessibility issues:
     a) Number of horizontal tasks that are delayed or giving up due to issues relating to accessibility.
     b) Users Comments
   - The number of accessibility issues in relation to horizontal interaction in the user comments.

7) Cross-Platform Understandability: it is the extent to which users can understand each service cross-platform easily in terms of the available data and functions through each service, as well as understanding the differences of user interfaces structure cross-platform. The following measurable criteria can be used to measure understandability:
   - Transparency: the user interfaces cross-platform needs to be transparent enough so that users can understand the level of data and functions available to them. The following are examples of metrics that can be used to measure transparency:
     a) The number of transparency issues in user comments for each service.
     b) The number of users who are trying to access unavailable data or functions after transition from a user interface to another.
   - Clarity: refers to clarity of the structures of services cross-platform, which need to be easily perceived by the mind and senses. The following is an example of a
metric that can used to measure clarity.

a) After transferring a task from a user interface to another, the number of horizontal tasks that are stopped by users due to the difficulty of understanding the structure of the transferred-to user interface.

8) Cross-platform Satisfaction: is the users’ perceptions, feelings, likeability and opinions on all the services cross-platform when attempting their goals horizontally. Measuring satisfaction cross-platform can be based on the following:

- Within-platform sub-task level satisfaction: asking questions immediately after a user attempts a sub-task within an individual platform before switching to another platform.
- Cross-platform task level satisfaction: asking questions immediately after a user attempts a horizontal task.
- Cross-platform test level satisfaction: asking questions after a user attempts all cross-platform tasks.

It is also important to mention that the System Usability Scale (SUS), which is a simple survey of Likert scales created by Brooke [46] in 1996 to quickly benchmark software usability, is not applicable to be used with cross-platform satisfaction. This is because SUS focuses on satisfaction when users use a single user interface. Our approach is to amend statements in the SUS to become Services Usability Scale (SUS), which can be used for measuring user satisfaction of services cross-platform. An example of a statement within the Service Usability Scale is “I thought there was too much inconsistency between services cross-platform”.

9) Cross-platform Universality: it is the level to which services cross-platform can provide consistent considerations of cultural backgrounds so they can be used by all kinds of users, for example, the language and cultural conventions need to be consistent in all user interfaces (consistent use of symbols and numeric formats), and concerning translation, we need to use consolidated language for labels cross-platform, and so on. The following measurable criteria can be used to measure universality cross-platform.

- Universality issues and horizontal tasks:
  a) The proportion of the time a user spends on recovering from universality issues when attempting a horizontal task.
  b) Number of users who stop completing horizontal tasks due to any universality issue.

- Universality consistency and horizontal tasks:
  a) Number of times a user encounters an issue relating to inconsistency of considerations for cultural backgrounds cross-platform.

10) Cross-Platform Helpfulness is the extent to which a user can get the help or guidance they need at any point during his or her horizontal interaction with multiple user interfaces. The measurable criteria for this factor are presented below:

- Findability
  a) The number of users who are able to find appropriate help or documentation of resources in a reasonable time at any point during their horizontal interaction.
  b) The number of users who are able to complete their tasks that have been stopped before accessing the help or documentation resources.

- User Comments.
  a) The number of issues relating to cross-platform helpfulness in users comments.

11) Cross-platform Safety: the use of a certain service should not contribute to harming users or their resources when using another service. For example, if a user uses a delete function within a specific service from a specific device for deleting a single object, the same function in other services, from other devices, should not be used to do another action (e.g., to delete more than one object). Also, user interfaces cross-platform should help users in understanding that performing a specific action within a specific platform can harm or damage their resources anywhere cross-platform (e.g., deleting a folder from device “A” can lead to deleting the same folder from other devices). In addition, user interfaces cross-platform need to be used safely in different contexts of use. For example, when a user transfers their interaction from a certain context of use to another (e.g., from standing to a moving context of use), the service needs to recognize this transition and the system behavior needs to be changed to reduce possible harm to users. This is because the manageability and attentiveness level of user interface elements can decrease in a moving context of use. As a consequence, undeliberate actions can occur such as pressing an unwanted link. Therefore, it might be useful to increase the safety of the user and their resources by using, for example, more error prevention strategies, such as confirmation messages. Measurable criterion, which can be used for user interfaces safety when interacting cross-platform, are listed below with its associated metrics:

- Safety of resources:
  a) Number of usability issues a user encounters in relation to the safety of their resources.
  b) The percentage of users who do any action that (not deliberately) affects their resources cross-device.
  c) Comparison of the number of usability issues a user encounters in relation to safety when using services in different contexts of use.

12) Cross-Platform Visibility is the capabilities of cross-platform services to provide coherence visibility of their status from any used service at hand. In other words, the cross-platform services need to communicate with each other to show users what is going on in each service through proper feedback and within a reasonable time. For example, if a user has synced a file from a PC to a mobile phone, the interactive system in a PC should show the user that the file has been synced to the mobile phone, so users are not required to check the mobile phone to see if the file has been synced. If the file has not been synced to the mobile phone due to, for example, memory capacity, the system in the PC should also give clear feedback to the user in regard to this. Also, if there is more than one mobile phone, the feedback needs to be specific and indicate to which mobile phone the file has been synced. Some measurable criteria of cross-platform
visibility are shown below:

- Visibility issues in user comments (e.g., the number of cross-platform visibility issues in the user comments).
- Task completeness and visibility issues (e.g., number of users who stop completing tasks due to the invisibility of their current status on any of the cross-platform services).
- Feedback (e.g., the number of users who express that they are still confused after reading the system feedback).

Factors and Other Measurable Criteria: There are some other measurable criteria, which can be associated with more than one factor. These measurable criteria are presented in the following Table VII, with the factors that they are associated with:

<table>
<thead>
<tr>
<th>Measurable Criteria</th>
<th>Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fluency</td>
<td>Efficiency, effectiveness, memorability, productivity, satisfaction.</td>
</tr>
<tr>
<td>Flexibility</td>
<td>Efficiency, effectiveness, satisfaction.</td>
</tr>
<tr>
<td>Consistency</td>
<td>Efficiency, effectiveness, learnability, memorability, productivity, understandability, satisfaction.</td>
</tr>
<tr>
<td>Controllability</td>
<td>Efficiency, effectiveness, safety, satisfaction.</td>
</tr>
<tr>
<td>Readability</td>
<td>Efficiency, effectiveness, learnability, memorability, productivity, helpfulness, accessibility, satisfaction.</td>
</tr>
<tr>
<td>Portability</td>
<td>Efficiency, effectiveness, productivity, satisfaction, safety.</td>
</tr>
<tr>
<td>Privacy</td>
<td>Safety, satisfaction.</td>
</tr>
<tr>
<td>Familiarity</td>
<td>Learnability, satisfaction.</td>
</tr>
<tr>
<td>Appropriateness</td>
<td>Learnability, satisfaction.</td>
</tr>
</tbody>
</table>

The measurable criteria from Table VII are described in detail below:

- **Fluency**: the capability of services to support the fluency of user activity cross-platform. For example, if a user transfers activity from a service in their PC to the service via a mobile phone, to what extent does the service in the mobile phone support fluency of the activity? There are different types of fluency where each one of these types can represent a measurable criterion, and has its own metrics. We consider task, content, interaction, cognition, and feeling fluency.

  - **Task fluency**: the capability of services in supporting fluency of a task when users are carrying out a horizontal task. For example, if a user has started on a task and reaches step 2 via service “X” in device “A” and decided to complete the task via device “B”, then the service “X” should support task fluency, for example, by allowing the user to continue on the task from step 2. The following are examples of possible measurements for task fluency:
    a) Comparison between the needed time to perform the task vertically (within individual platform) and horizontally (cross-platform).
    b) Time taken by users to resume the task after transition, which he or she has already started within another platform.

  - **Content fluency**: the capability of cross-platform services to support continuity of reading or exploring the contents after switching from one platform to another. For example, if a user was reading content using service “X” in device “A”, and has reached the last paragraph for the content, the service “X” via device B should support users to complete the reading easily and efficiently by, for example, showing the user the last paragraph immediately when accessing the service via device B. The following are examples of possible measurements for content fluency:
    a) After switching from a user interface “A” to user interface “B”, the time taken by a user to find the recent read or explored content via user interface “A” within user interface “B”.
    b) The percentage of users who are able to reach the same content after transition.

  - **Interaction fluency**: the capability of services in supporting fluency of user interaction when users are carrying out a task cross-platform. For example, if a user has started filling an online form via service “X” in device “A” and decided to complete filling the form via device “B”, then service “X” should support user interactions, for example, by auto-filling the fields in device “B” that the user has already filled via device “A”. We can measure this, for example, with the number of tasks a user needs to restart when working on it from the beginning after transition.

  - **Cognition fluency**: the capability of services in helping users to remain in the same level of engagement after transition of the activity to another device. A possible metric is as follow:
    a) After transition between devices, the time taken by users before a user recommences interacting with the transferred-to user interface.
    b) After transition between devices, the number of correct actions by users progressing towards the task goal.

  - **Feeling fluency**: the capability of services in helping users to remain in the same level of feeling after transition. One possible way of measuring these criteria is based on the number of users who express that they are unsatisfied or do not like (or the opposite: more satisfied or like) the current user interface in comparison to the previous one which a user transferred his or her task from to the current one.

    - **Flexibility**: the number of possible pathways in each service that the users are able to take for the purpose of continuing a task after migrating it to another device. For example, assume that a user reads a book using a user interface in device “A”, and reaches page 332, then the user decides to complete the reading using the user interface in device “B”. Then, what are the possible pathways that the user interface in device “B” can offer for this user to go immediately to page 332 in device “B”? We can measure this measurable criterion based on the maximum number of pathways that users have taken to reach a specific target.

    - **Consistency**: various levels of consistency to which cross-platform services can provide, including: a) the structure of information, and location of elements need to be consistent in all services cross-platform, so users can build a consolidated perception of the entire interactive system cross-platform, b) the label of user interface objects needs to be the same, as much as is possible, cross-platform, c) the number of operations or steps to achieve a specific task need to be the same cross-platform, d) the design patterns need to be the same as possible cross-platform, e) the data, functions
and features cross-platform need to be the same as much as possible, f) appearance and aesthetic design need to be consistent. Consistency can be measured based on the number of users who fail to achieve a task or sub-task due to the inconsistency of services. The number of usability issues identified in relation to consistency cross-platform is another possible metric.

- Controllability: the extent to which users can feel in control over an individual service within the specific platform and over all the services cross-platform.

**Controllability within the individual platform:** the capability of a service within a specific platform to allow users to feel in control so the results of their interactions cannot affect the service that can be accessed via other devices (e.g., downloading a large file from a PC should not result in downloading the file to the mobile device). This can be measured based on the number of usability issues encountered by users in relation to controllability within the individual platform.

**Controllability of the service cross-platform:** the capability of cross-platform services that allows users to feel in control over all the services (e.g., sync files to specific services within specific devices). In addition, the capability of cross-platform services to support reversibility of user action from any device (e.g., from device “B”, a user can undo the command “deleting a word” that has been performed from device “A”, so that it is not required for the user to go to device A to do the command). Controllability cross-platform can be measured based on the number of usability issues encountered by users in relation to controllability cross-platform.

- Readability: the visual content in all services cross-platform are easy to understand, as well as being consistent across all services. It can be measured based on the number of issues related to readability that might be encountered by users when performing cross-platform tasks.

- Portability: the capability of services in supporting a user in carrying out a task cross-platform with considerations of the portability factors that may affect usability (e.g., considering using one hand in different contexts of use [i.e., moving or driving]). Possible metrics for this criteria are below:

a) The number of usability issues that can be identified within the specific context of use in comparison to other contexts of use.

b) The number of usability errors that a user may make when attempting a task in a specified context of use in comparison to other contexts of use.

- Privacy: the level to which the cross-platform services can help user understanding that their data, and especially sensitive data, cannot be revealed to others. In addition, the capability of the services to support protection of user sensitive data in different contexts of use (e.g., if a user sets his or her preference as “show password characters” from service “X” via PC, this setting should not be applied when using service “X” from the mobile device in a rich-sensory environment), and password characters should automatically be set to hidden. The number of privacy issues that can be reported by users in different contexts of use can be used to measure privacy criteria.

- Familiarity: the level to which an individual service within a specific platform can provide familiar elements to users based on their previous experience of interacting with the same service from another platform. The familiarity criterion can be measured based on the user comments in relation to encountered unfamiliar elements in any service cross-platform.

- Appropriateness: the capabilities to provide appropriate consistent metaphors in each service cross-platform. Appropriateness can be measured, for example, from the number of user comments concerning inappropriate or inconsistent use of metaphors cross-platform.

Using the CPUM model requires observing users when they interact with multiple user interfaces to achieve a horizontal goal. In the following section, we present the cross-platform usability test procedures.

**V. PROPOSED USABILITY TEST PROCEDURE OF CROSS-PLATFORM SERVICES**

Usability tests need to be planned before conducting usability testing sessions. Fig. 4 illustrates a simple way of using cross-platform usability test procedures.

![Cross-platform usability test procedure](image)

The procedural steps are described below:

1) Screening the questionnaire that needs to be used to determine the right participants for the test,

2) Greeting, welcoming and signing consent form and non-disclosure agreements,

3) A pre-questionnaire is used to gain some historical
information about the participants,
4) Allowing participants to practice thinking aloud protocols, and familiarizing themselves with all user interfaces cross-platform that will be tested,
5) Participant conducts horizontal tasks (including n number of sub-tasks to be performed with n services).
6) Testing the user satisfaction at three stages (after each sub task, after each horizontal task and after all horizontal tasks).
7) Debriefing and garnering the thoughts of participants.

The procedural steps presented here are different from the procedures that are followed traditionally when testing an individual service within a specific platform. The cross-platform usability test procedures involve more than one interaction session for each participant because each participant needs to perform horizontal tasks using more than one device.

VI. CONCLUSIONS

Users can currently achieve their goals horizontally using multiple user interfaces. The current models for designing usable systems are developed with consideration of only vertical interactions. We have developed and presented a cross-platform measurement model that encompasses 12 usability factors with their measurable criteria and metrics. Test procedures that can be used in association with the model are also proposed.

REFERENCES

[45] G. Óquist, M. Goldstein, and D. Chincholle, “Assessing usability across multiple user interfaces,” Multiple User Interfaces:
Khalid Majrashi was born in Jazan, Saudi Arabia, in 1987. He received the B.E degree in education of computer science from the University of Jazan, Jazan Saudi Arabia, in 2008, and the master degree in computer science from RMIT University, Melbourne, Australia in 2012, and currently he is a PhD candidate (in the field of computer science) at RMIT University, Melbourne, Australia.

In 2012, he joined the Department of Computer, Institute of Public Administration, Riyadh, Saudi Arabia, as a lecturer. He has published a book in combination with Margaret Hamilton entitled “User Experience of University Websites” (Saarbrücken Germany, LAP LAMBERT Academic Publishing, 2014). His current research interests include software quality, user experience, usability, mobile computing, cloud computing (software as a service), and user interface models.

Margaret Hamilton received BSc (Hons), PhD, from University of Wollongong, Australia. She is an associate professor in the School of Computer Science and Information Technology at RMIT University in Melbourne Australia. Her current research centres on mobility, human computer interaction and computer science education.