Software Qual J (2017) 25:743–795
DOI 10.1007/s11219-016-9320-z

Quality characteristics and measures for human–computer interaction evaluation in ubiquitous systems

Rainara Maia Carvalho[1] · Rossana Maria de Castro Andrade[1] · Káthia Marçal de Oliveira[2] · Ismayle de Sousa Santos[1] · Carla Ilane Moreira Bezerra[1]

Published online: 7 July 2016

Abstract The advent of ubiquitous systems places even more focus on users, since these systems must support their daily activities in such a transparent way that does not disturb them. Thus, much more attention should be provided to human–computer interaction (HCI) and, as a consequence, to its quality. Dealing with quality issues implies first the identification of the quality characteristics that should be achieved and, then, which software measures should be used to evaluate them in a target system. Therefore, this work aims to identify what quality characteristics and measures have been used for the HCI evaluation of ubiquitous systems. In order to achieve our goal, we performed a large literature review, using a systematic mapping study, and we present our results in this paper. We identified 41 pertinent papers that were deeply analyzed to extract quality characteristics and software measures. We found 186 quality characteristics, but since there were divergences on their definitions and duplicated characteristics, an analysis of synonyms by peer review based on the equivalence of definitions was also done. This analysis allowed us to define a final suitable set composed of 27 quality characteristics, where 21 are generic to any

✉ Rainara Maia Carvalho
rainaracarvalho@great.ufc.br

Rossana Maria de Castro Andrade
rossana@great.ufc.br

Káthia Marçal de Oliveira
kathia.oliveira@univ-valenciennes.fr

Ismayle de Sousa Santos
ismaylesantos@great.ufc.br

Carla Ilane Moreira Bezerra
carlabezerra@great.ufc.br

[1] Group of Computer Networks, Software Engineering, and Systems (GREat), Computer Science Department (DC), Federal University of Ceará (UFC), Fortaleza, Brazil

[2] Laboratory of Automatic Control, Mechanics and Computer Science for Industrial and Human-Machine Systems (LAMIH), CNRS UMR 8201, University of Valenciennes and Hainaut-Cambrésis (UVHC), Valenciennes, France

system but are particularized for ubiquitous applications and 6 are specific for this domain. We also found 218 citations of measures associated with the characteristics, although the majority of them are simple definitions with no detail about their measurement functions. Our results provide not only an overview of this area to guide researchers in directing their efforts but also it can help practitioners in evaluating ubiquitous systems using these measures.

Keywords Ubiquitous systems · Human–computer interaction · Quality model · Quality characteristics · Software measures · Systematic mapping study

1 Introduction

The increasing improvement in the miniaturization of computational devices and in the wireless communications has been an important factor for advances in the ubiquitous systems development. These systems change the focus of interest from computer technology to users and their needs (Rocha et al. 2011). They are capable of monitoring users and their environments to provide relevant services in a transparent and intuitive way, changing completely the way users interact with systems. This suggests new challenges for human–computer interaction (HCI) evaluation in ubiquitous systems.

These issues are even more relevant if we consider that ubiquitous systems are present anywhere and anytime for users, which leads to a high risk of users feeling annoyed and overwhelmed by ubiquitous systems. No user would like to have several systems requiring too much interruption with irrelevant information everywhere and any time (Evers et al. 2014). Considering this scenario, we argue that ubiquitous systems should be delivered to the user by prioritizing the quality of interaction. Therefore, to properly execute an HCI quality evaluation in ubiquitous systems, it is essential to know which quality characteristics and measures have to be taken into account.

Looking to the international standard Software Quality Requirements and Evaluation (SQuaRE) for any type of system (ISO/IEC 25010 2011), several quality characteristics could be considered to evaluate HCI in ubiquitous systems (e.g., *Usability*, *Freedom from Risk* and *Context Coverage*), and also several software measures could be used to evaluate these characteristics. However, considering that ubiquitous systems present a particular type of interaction, which is the natural and transparent interaction (Poppe et al. 2007), and particular characteristics, like Context-Awareness and Adaptability (Evers et al. 2014), we believe that new specific characteristics and measures could be also applied in an HCI quality assessment.

Then, we conducted a systematic mapping (SM) study (Petersen et al. 2008) to identify which quality characteristics and measures should be taken into account for ubiquitous systems. The SM is a research method that provides a broad overview of a research area to establish whether research evidence exists on a topic and whether it provides an indication of the evidence quantity (Kitchenham and Charters 2007). Based on the SM study, this paper presents the set of quality characteristics and measures that one should consider while performing an HCI quality evaluation of ubiquitous systems. It is important to point out that we did not find a work like ours that aggregates characteristics and measures for evaluating ubiquitous systems from an extensive literature review and organizes them using a standard quality model.

The remainder of this paper is organized in five sections. Section 2 describes an overview of ubiquitous systems, emphasizing HCI issues. Section 3 describes the research method we used, and Sect. 4 presents the obtained results. In Sect. 5, the results are discussed through a classification of the final set of the characteristics according to SQuaRE quality models. Section 6 presents the threats to validity of the study. Section 7 presents related work and a comparison with our study. Finally, Sect. 8 presents our conclusions and future work.

2 Background

Mark Weiser's vision of ubiquitous computing is well expressed in his following famous quote: "The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it" (Weiser 1991). Therefore, this paradigm includes services and the provision of information to support users in everyday tasks by a variety of computers. Moreover, this support should be executed without users noticing that they are interacting with several technologies.

To achieve this vision, the system should be able to understand the user's behavior and adapt itself. This is enabled by the context-awareness characteristic, which captures relevant information during the interaction between users and applications, and applies it to support users in performing their tasks (Dey 2001). This characteristic allows the system to know, for example, who the user is, where he/she is, what he/she is doing in a given time, and what makes it possible to deliver several relevant services to the user.

Adapting the HCI evaluation of ubiquitous systems is even more relevant to this scenario if we consider the following four differences between interaction in traditional systems and in ubiquitous systems, according to (Poppe et al. 2007):

- *New possibilities of sensing* In traditional systems the inputs of the users are provided often by hardware devices, such as keyboard or mouse. In ubiquitous systems, inputs can be captured by sensors (e.g., GPS, accelerometer and magnetometer) without the user noticing or captured by the voice, gesture and touch. These new sources of inputs make the interaction more natural.

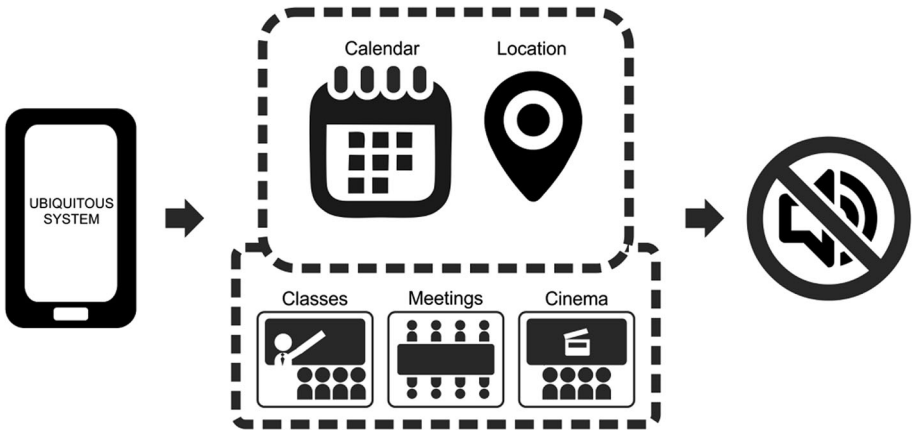


Fig. 1 Scenarios which a ubiquitous application can help

- *Shift in initiative* In traditional systems, the HCI corresponds to an explicit dialogue between the user and the computer, and usually, it is the user who begins the interaction. In ubiquitous systems, dialogues can be initiated by the system itself, given its ability to sense the user, his/her environment and his/her needs.
- *Heterogeneity of physical interfaces* Ubiquitous systems can be present in several everyday objects. Thus, there is a movement to make ubiquitous systems for both large interfaces, like interactive display, and small ones, like smartphones and wearable devices.
- *Shift in application purpose* Ubiquitous systems focus on the user and on everyday life, whereas traditional systems are, in general, task-based.

In Fig. 1, the ubiquitous system puts a mobile phone in silent mode when the presence of the user in an event like meeting or cinema is detected. This system does not use traditional input devices such as keyboard or mouse, but inputs (e.g., activity and location) captured by both physical and logical sensors (e.g., GPS for location and Calendar for activity) without users' perception. Besides that, in this example, the interaction is initiated by the system; the user does not have to take actions. Instead of this, the system substitutes an action usually performed by the user (e.g., put a mobile phone in silent mode).

Besides these differences, Bezerra et al. (2014) mention three challenges for usability testing in ubiquitous systems:

- Ubiquitous environments have more usability factors that should be evaluated, such as contextual information. Thus, it is necessary to predict all relevant changes in context and analyze when those changes can impact on the behavior of the system;
- Most of the software measures do not consider the factors of ubiquitous applications. It is a challenge to make the evaluation of the usability of these systems more reliable and to identify measures that consider the ubiquitous features in usability tests; and
- Currently, usability testing methods follow the same activities performed in traditional systems. Research need to be conducted to elaborate an approach for usability testing with specific tasks and measures to evaluate ubiquitous systems.

Based on all these particularities of HCI in ubiquitous systems and the challenges mentioned before, we were convinced that, for an adequate quality assessment of ubiquitous applications, we need a deep analysis of specific quality characteristics and measures for this type of application and to achieve that we need to first investigate the existing characteristics and measures that have been explored in the literature.

3 The research method: systematic mapping

Systematic mapping (SM) is a method to build a classification scheme and to structure a field of interest (Petersen et al. 2008). It is defined as a rigorous, unbiased and auditable procedure for searching research literature. Systematic mapping studies use the same basic methodology as systematic review (Kitchenham et al. 2010) guided by research questions. Nevertheless, the research questions for a mapping study are more general, related to research trend, and quite high level, including issues such as: Which subtopics have been addressed, what empirical methods have been used, and what subtopics have sufficient studies for a more detailed system review.

To perform our systematic mapping, we followed a process with three main activities proposed by Kitchenham and Charters (2007) for systematic studies: (1) planning; (2) conducting; and (3) reporting (see Fig. 2). The definition of steps for each activity was

based on Petersen et al. (2008), Silveira et al. (2011) and Wohlin (2014). The first activity (*planning)* aims to define the protocol that will guide all research. The second activity (*conducting)* aims to execute the defined protocol. In our study, this activity was performed in two different phases. In the first one, the primary studies selection was based on database search, i.e., we used digital libraries to start our search. In the second one, we use snowballing procedures defined by Wohlin (2014) in order to complement the set of papers found by the database search, as performed by Tahir and Jafar (2011).

3.1 Planning: definition of protocol

The aim of the planning phase is the definition of a review protocol (single step in this phase as shown in Fig. 2). This protocol is composed of the following information:

A. *Research Questions* The aim of our study was to identify the quality characteristics and measures for HCI evaluation of ubiquitous systems. Therefore, we established the following research questions:

RQ1 What quality characteristics have been proposed for ubiquitous systems' HCI evaluation?

RQ2 What software measures have been proposed for ubiquitous systems' HCI evaluation?

Knowing that, usually, quality characteristics are organized in a hierarchical tree [named a quality model (ISO/IEC 25000 2014)] that goes from a generic definition up to

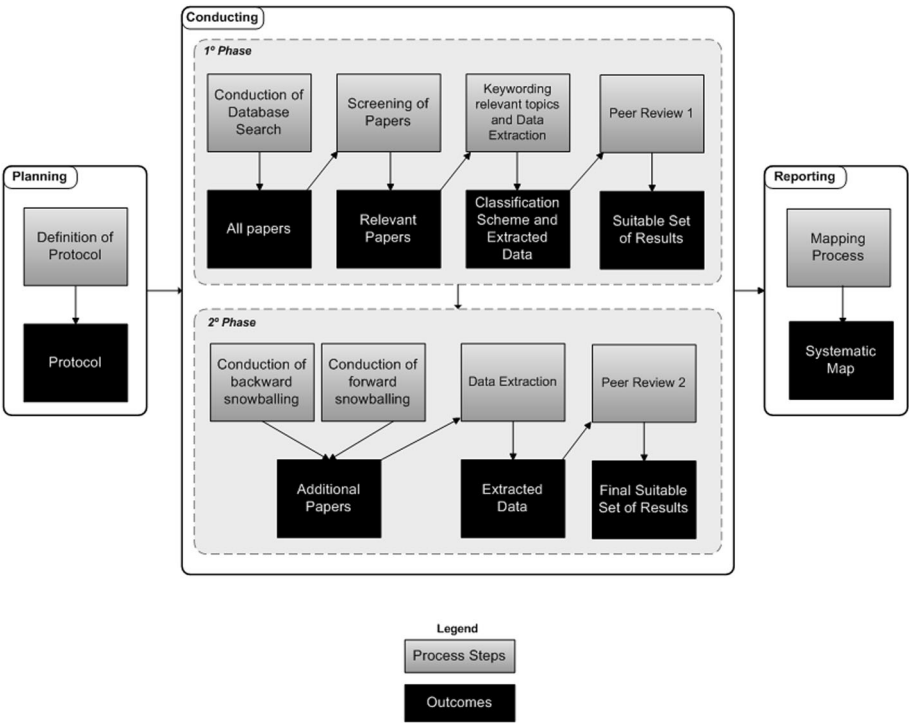


Fig. 2 The systematic mapping process

measures that allow the product assessment, we established a third research questions as follows:

RQ3 Are the characteristics and measures organized in quality models for ubiquitous systems' HCI evaluation?

B. *Key terms* The key terms were derived based on the research questions, identifying the object we were looking for (quality characteristic, measures and quality models), the purpose (HCI evaluation) and the context (ubiquitous systems). After that, some synonyms or alternative words were added in the set of key terms. When analyzing the HCI field, standards and research usually talk about assessment, methods and techniques related to usability (see, e.g., Nielsen 1994; ISO 9241-11 1998; Sears and Jacko 2009). In this way, we include usability evaluation as another term. The final key terms are presented in Table 1.

Regarding the term "ubiquitous systems," we avoided the use of the term "systems," because we did not want to limit the search for any kind of software. Furthermore, we agree with Petersen et al. (2008) that adding specific outcomes is a restriction and the mapping study aims to have a broad overview of the research area as a whole. If we had only considered certain types of software (e.g., "systems," "applications," "services") the overview could have been biased and the map incomplete.

We also avoid to use the term "mobile" as a synonym of ubiquitous since not all mobile applications are ubiquitous applications. As the main goal of this research is to find characteristics only for ubiquitous systems, we believe that if we had added mobile in our string, several irrelevant papers to our research questions could appear. Other SLR studies (Spínola and Travassos 2012; Viana et al. 2014) also do not consider the keyword "mobile" as synonym of ubiquitous.

C. *Search String* Based on the key terms previously presented, the following search string was defined:
((characteristic OR measure OR metric OR "quality model" OR framework) AND (HCI OR "human–computer interaction" OR "human–computer interface" OR "user interface" OR interaction OR usability) AND (evaluation OR assessment) AND (ubiquitous OR pervasive))

We used three control papers (Scholtz and Consolvo 2004; Kim et al. 2008; Song et al. 2009), which means papers that we expected to appear in the results, because we already

Table 1 Key terms

Object	Purpose	Context
Characteristic	HCI evaluation	Ubiquitous
Measure	Human–computer interaction evaluation	Pervasive
Metric	Human–computer interface evaluation	
Quality model	User interface evaluation	
Framework	Interaction evaluation	
	Usability evaluation	
	HCI assessment	
	Human–computer interaction assessment	
	Human–computer interface assessment	
	User interface assessment	
	Interaction assessment	
	Usability assessment	

know they answer our research questions. Once they were present in the results, it indicated that the search string was validated to execute the systematic mapping.

D. *Research Sources* To obtain the primary studies, we used two kinds of search: database search and snowballing. For the database search, we selected the most relevant digital libraries used in other systematic studies: ACM Digital Library (http://dl.acm.org/), IEEE Xplore (http://ieeexplore.ieee.org/), Scopus (http://www.scopus.com/scopus/home.url), Science Direct (http://www.sciencedirect.com), SpringerLink (http://www.springer.com/) and Compendex (http://www.engineeringvillage.com/). For the snowballing, we used the backward (i.e., checking the references list of the studies) and forward (*i.e.*, checking papers that cited the studies) procedures. For the forward snowballing, we used Google Scholar for a broader search since the search engines from these databases, such as Scopus and IEEE, limit the search only for papers indexed from them. Furthermore, Wohlin (2014) suggests uses Google Scholar for forward snowballing procedures.

E. *Study Selection Criteria:* We have defined the following selection criteria in order to select the most suitable studies:

SC1—The study should be written in English;

SC2—The study should be dated from 1991 or later. We choose this date because the term "ubiquitous computing" appeared in the paper of Weiser (1991), considered the father of ubiquitous computing;

SC3—The study should be available in the internet that means even if not available directly in the digital library, it should be possible to find it by internet facilities;

SC4—The study should present initiatives related to HCI evaluation on ubiquitous applications (no other contexts like desktop, web systems or HCI development); and

SC5—When the same study was published in different papers, only the most complete and recent was included, as suggested by Silveira et al. (2011) in systematic mappings.

It is important to highlight that no restriction was defined for the kind of paper selection that means all kinds of study (papers in conference or in journal, books, book chapters, short and long papers, etc.) were accepted. They were processed in the same way considering the above selection criteria.

3.2 Conducting

As presented in Fig. 2, this activity was performed in two phases. The first one is composed of four steps: (1) Conduction of database search, which is performed to find relevant papers in digital libraries using well-defined search strings; (2) Screening of papers; (3) Keyword relevant topics and Data extraction; and (4) Peer review 1.

The second phase is composed of four steps: (1) Conduction of backward snowballing, which implies seeking papers from reference lists of the identified papers in the conducting activity's first phase, (2) Conduction of forward snowballing, which implies seeking papers that have cited the papers found in the conducting activity's first phase, (3) Data extraction and (4) Peer review 2. All these steps are described in the next subsections.

3.2.1 Conducting: first phase

3.2.1.1 Conduction of database search In this step, we searched papers based on the defined protocol. The selection was done on April 9, 2013. The set of search strings was applied within the search engines (*ACM*, *IEEE*, *Scopus*, *Compendex*, *Springer and Science*

Direct), and all information about the papers, including title and abstracts,[1] was downloaded and imported to the Start tool[2] (Hernandes et al. 2012), which is a free tool that supports systematic review's activities. This tool was selected because it is free, easy to use, and the authors have experience in using it. This step retrieved 1170 papers (see Fig. 3), 500 from Compendex (42.7 %), 269 from Scopus (23 %), 268 from Springer (22.9 %), 101 from IEEE (8.6 %), 24 from ACM (2.1 %) and 8 from Science Direct (0.7 %).

3.2.1.2 Screening of papers This step involved the selection of studies considering three filters, described in Fig. 4. The aim of the first filter was to exclude duplicated papers, because some papers appeared in several sources, and thus, just one of them was included. We identified 302 duplicated papers (26 %) in the initial set of 1170 papers. Thus, 868 remaining papers (74 %) were selected to the next filter.

The aim of the second filter was to apply the defined selection criteria reading the abstract and title. This analysis was performed by peers, because we would like to avoid bias in the selection process. Thus, one researcher reviewed the selection of the other. To that end, we performed several face meetings during 1 week, where one peer reviewed the selection of the other, and, in case of disagreement, we opened a discussion to reach a consensus. Although this process could seem long, it was better to all peers since they previously schedule the meeting in their agenda to work on the selection process. We rejected 749 papers (86.30 %) and accepted 119 papers (13.70 %).

To apply the third filter, we downloaded the 119 papers and performed a detailed reading. This step was performed by four researchers. Two other researchers participated in the review of the papers that caused some doubt. The selection criteria were applied once again. As a result, we got 87 rejected papers (73 %) and 32 accepted papers (27 %). From the total of rejected papers, 85 were eliminated by the selection criterion **SC4** and 2 papers by **SC5**.

The following 32 accepted papers went to next phase (Keyword relevant topics and Data extraction): (Abi-Char et al. 2010; Cappiello et al. 2009; Chang and Lin 2011; Damián-Reyes et al. 2011; De Moor et al. 2010; Evers et al. 2010; Haapalainen et al. 2010; Iqbal et al. 2005; Jafari et al. 2010; Jia et al. 2009; Kemp et al. 2008; Kim et al. 2008; Ko et al. 2010; Kourouthanassis et al. 2008; Kryvinska et al. 2011; Lee and Yun 2012; Lee et al. 2008; Liampotis et al. 2009; Ranganathan et al. 2005; Ross and Burnett 2001; Rubio and Bozo 2007; Schalkwyk et al. 2010; Scholtz and Consolvo 2004; Sousa et al. 2011; Sun and Denko 2008; Thompson and Azvine 2004; Toch 2011; Wagner et al. 2012; Waibel et al. 2010; Weihong-Guo et al. 2008; Wu and Fu 2012; Zhang et al. 2006).

3.2.1.3 Keyword relevant topics and Data extraction To finish the *conducting review phase*, we should define the classification scheme, which will serve to create our systematic map (the main result of an SM). This classification scheme is composed of at least two facets. It is defined by keywording relevant topics in the abstract of the papers, which means the searching of keywords and concepts that reflect the contribution of the study. Reading the papers, we identified concepts that reflect the main following contributions: definition of the quality model, conceptual frameworks of measures and quality

[1] Some papers did not have the abstract registered in the database. For those papers we downloaded the complete paper to start the review (next step screening the papers).

[2] http://lapes.dc.ufscar.br/tools/start_tool.

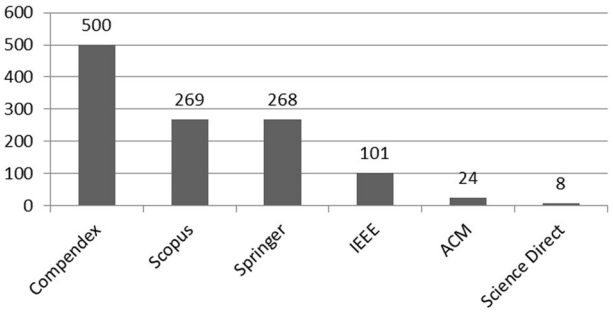
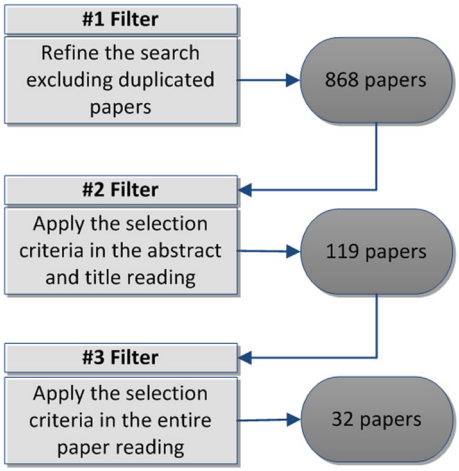


Fig. 3 Amount of studies versus sources

Fig. 4 Screening process



characteristics. Therefore, we defined the contribution type as one facet in our classification scheme, as shown in Table 2.

Furthermore, we decided to also use the research type of (Wieringa et al. 2005) as a second facet (see Table 3). This facet is suggested by Petersen et al. (2008) since it reflects the research approach used in the papers and it is independent from a specific topic.

Finally, we performed the Data extraction from each selected paper. To that end, the necessary information to answer each research question was extracted, meaning that the paper presented quality characteristics (answering the research question 1), software measures (answering the research question 2) and/or quality model (answering the research question 3). Besides that, it was necessary to classify each paper considering the defined classification scheme. Table 4 presents the Data extraction form used in our study.

3.2.1.4 Peer review 1 After the Data extraction, we have noticed that several of characteristics have the same meaning, but are presented with different names. To obtain a suitable set of quality characteristics, we performed an analysis by peer review (see Fig. 5), considering what was described in each paper. Three peer reviewers participated in the process. This analysis was performed in three steps. First, one researcher read all papers and identified the characteristics and their synonyms. Second, characteristics and their

Table 2 Contribution type facet

Category	Description
Quality model	The paper presents a model as defined by the SQuaRE standard (ISO/IEC 25000 2014) in a hierarchical way (characteristics, subcharacteristics and measures). The measurements are presented
Characteristic framework	The paper organizes characteristics and/or subcharacteristics as a list of issues that should be evaluated in a ubiquitous system. No measure is presented
Measure framework	The paper presents measures organized into quality characteristics
Quality issues	The paper presents issues that should be considered in an evaluation of a ubiquitous system, but they are not explicitly defined as quality characteristics

Table 3 Research type facet. (*Source*: Petersen et al. (2008) and Wieringa et al. (2005))

Category	Description
Validation research	Techniques investigated are novel and have not yet been implemented in practice. Techniques used are, for example, experiments, i.e., work done in the laboratory
Evaluation research	Techniques are implemented in practice, and an evaluation of the technique is conducted. That means it is shown how the technique is implemented in practice (solution implementation) and what are the consequences of the implementation in terms of benefits and drawbacks (implementation evaluation). This also includes identifying problems in industry
Solution proposal	A solution for a problem is proposed, and the solution can be either novel or a significant extension of an existing technique. The potential benefits and the applicability of the solution are shown by a small example or a good line of argumentation
Philosophical papers	These papers sketch a new way of looking at existing things by structuring the field in a form of taxonomy or conceptual framework
Opinion papers	These papers express the personal opinion of somebody whether a certain technique is good or bad, or how things should been done. They do not rely on related work and research methodologies
Experience papers	Experience papers explain on what and how something has been done in practice. It has to be the personal experience of the author

original definition from the papers were organized in a document that was peer reviewed by two other researchers. For the peer review, the researchers identified if they agreed with the synonym identification or not. If they disagreed, they should write down some justification and propose a new organization. Third, a meeting was held for the final consensus.

3.2.2 Conducting: second phase

3.2.2.1 Conduction of backward and forward snowballing The snowballing procedure is usually performed with a start set of papers. In our study, the start set corresponds to the 32 papers found by the first phase of the conducting activity. The search was done on January 26, 2016. As previously described, we performed this phase after extracting and analyzing the first set of papers. In this way, we had a consistent and well-defined set of papers to consider as start set for the snowballing.

Moreover, we performed two types of snowballing to find additional papers: backward and forward. Figure 6 and 7 present the detailed procedure to search additional papers by

Table 4 Data extraction form

Quality characteristics	*Answering what quality characteristics are presented in the paper. Quality characteristics are desirable abilities in the system, for example, Usability and Context-awareness*
Software measure	*Answering what software measures are presented in the paper, for example, mean time taken to learn to use a function correctly*
Quality model	*Answering whether the paper proposes a structure composed of characteristics, subcharacteristics and measures according to the quality model defined by the SQuaRE standard. For more information about quality models, readers are referred to ISO/IEC 25000* (2014). () Yes () No
Application domain	*If the paper presents experimental studies, i.e., case studies, controlled experiments, answering what application domain is used to do that, for example, tour guides or smart house*
Research type	( ) Evaluation Research () Experience Papers ( ) Solution Proposal () Opinion Papers ( ) Philosophical Papers () Validation Research
Contribution type	( ) Characteristic Framework () Measure Framework ( ) Quality Issues () Quality Model

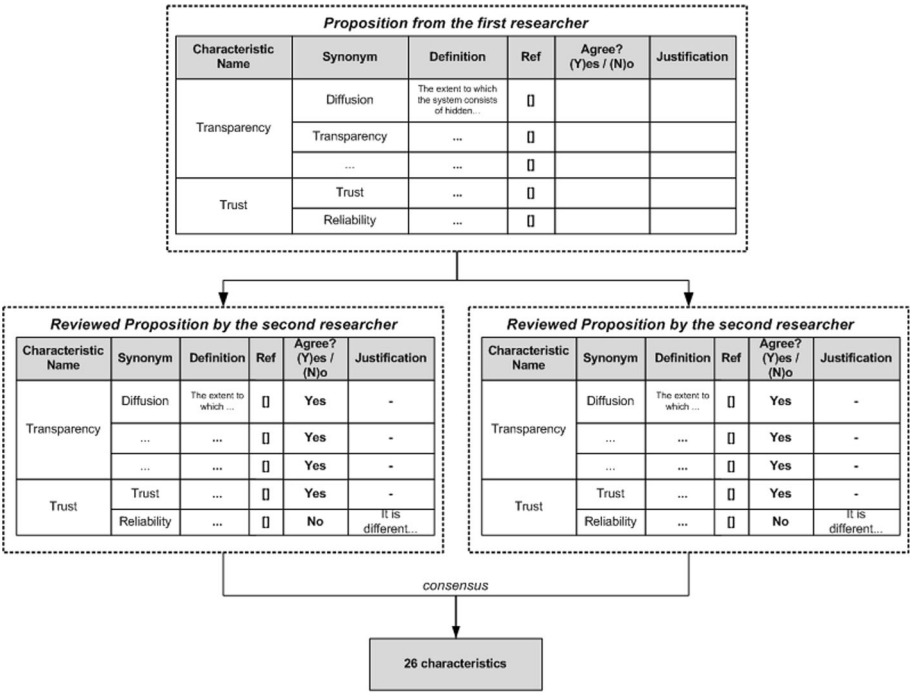


Fig. 5 Peer review 1

backward snowballing and forward snowballing, respectively. This procedure is adapted from the snowballing process proposed by Wohlin (2014).

For the backward snowballing procedure (Fig. 6), we start by listing all references from the 32 selected papers. We obtained 969 references in total. Then, we applied the two basic

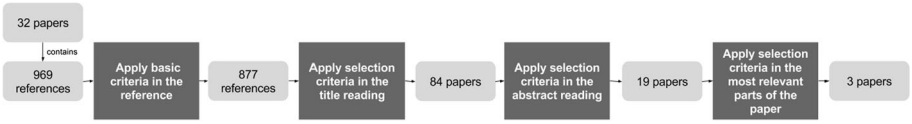


Fig. 6 Backward snowballing procedure

criteria from the protocol (i.e., SC1—papers written in English and SC2—published after 1991) and we excluded all references that were only web address of research groups, newspaper and/or companies. We reduce the set of references to 877 studies.

Next, we applied the other selection criteria (i.e., SC4: The study should present initiatives related to HCI evaluation on ubiquitous applications (no other contexts like desktop, web systems or HCI development)) by reading the title. Duplicated studies were also excluded. From 877 references, 793 references were excluded, which resulted in 84 studies selected for the abstract reading. By applying the SC4 criteria in the abstract reading, only 19 were selected. Then, the fourth step consisted of applying again the SC4 criteria by reading the most relevant parts of the paper. It is important to mention that it is not recommended to start reading the entire paper before Data extraction; instead, Wohlin (2014) recommends to browse through the paper and read the most relevant parts to make a decision in an efficient way. Following this idea, we browsed the paper looking the most relevant parts that means those that explained the quality characteristics, the software measures and the kind of system being evaluated. At the end, we obtained three papers for Data extraction.

For the forward snowballing, each one of the 32 papers was analyzed based on its citations. We obtained 962 papers that cite at least one of the 32 papers. The same procedure of backward snowballing was performed to the forward snowballing. At the end, 6 papers were selected for Data extraction.

As result of these procedures we obtained 9 papers, 3 from backward snowballing (Chalmers and Sloman 1999; Kim and Lee 2006; Ryu et al. 2006) and 6 from forward snowballing (Karaiskos et al. 2009; Karvonen and Kujala 2014; Sanchez-pi and Carb 2012; Jafari et al. 2011; Santos et al. 2013; Carvalho et al. 2015).

3.2.2.2 Data extraction The Data extraction was performed using the same Data extraction form used in the first phase of the conducting activity (see Table 4). The data from the nine papers were extracted by three reviewers. This extraction considered only quality characteristics related to the user interaction that means characteristics that impact the quality of HCI. For example, Ryu et al. (2006) propose characteristics for evaluating ubiquitous systems and middleware. In this work, there are characteristics related to internal quality of the system, but we extracted only characteristics related to HCI and not those related to internal quality. At the end, we obtained 52 quality characteristics to be analyzed.

3.2.2.3 Peer review 2 After the Data extraction of the papers from snowballing, a new peer review was performed in order to integrate the new extracted data to the data from the first phase of the conducting activity. The same three reviewers from the first peer review participated in the process in order to ensure consistent results. Like in the first peer review, initially, one researcher read all characteristics identified and its definitions and proposed integration with the set of existing characteristics. This information was organized in a

document that was peer reviewed by the two other researchers (see Fig. 8). This document presented the quality characteristic identified in the snowballing procedures and its synonyms, when pertinent, with one of the 26 quality characteristics identified in the previous peer review. In this peer review, the researchers identified if they agreed with the proposition or not. If they disagreed, they should write down some justification and propose a new organization. Finally, a meeting was held for discuss the divergences and getting a consensus.

The consensus obtained the following results: From the 52 characteristics of the Data extraction, 40 characteristics were integrated with the existing set of 26 characteristics, 11 characteristics were excluded because they were considered either not pertinent for ubiquitous systems or for HCI; 1 characteristic was considered as a new one. As a result,

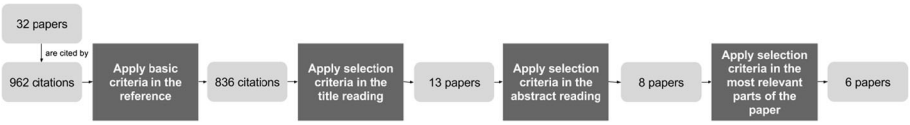


Fig. 7 Forward snowballing procedure

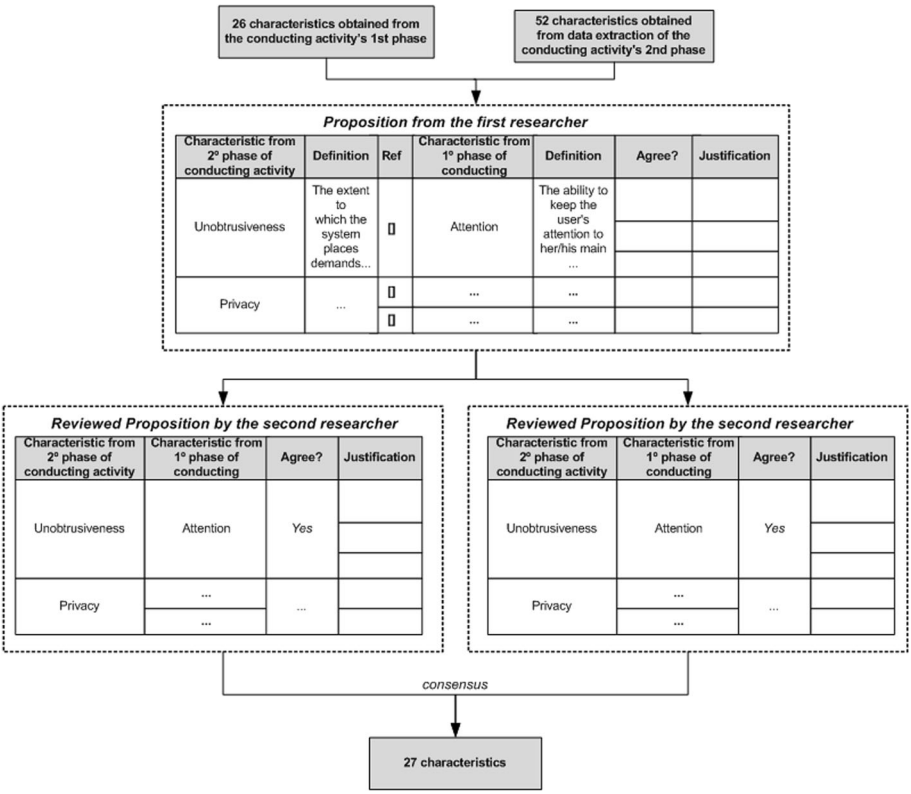


Fig. 8 Peer review 2

we obtained a final suitable set of 27 quality characteristics for HCI evaluation in ubiquitous systems.

4 Results of the systematic mapping

This systematic mapping found 41 papers that answer the defined research questions. All of them present quality characteristics for evaluating ubiquitous systems. Only 23 papers present software measures to evaluate the proposed characteristics, and 3 papers present a model composed of characteristics similar to the SQuaRE standard ISO/IEC 25000 (2014). Moreover, these papers cover different types of ubiquitous systems [such as, mobile applications (Cappiello et al. 2009; Chang and Lin 2011; Damián-Reyes et al. 2011; De Moor et al. 2010)], smart homes (Liampotis et al. 2009; Wu and Fu 2012), whiteboards (Scholtz and Consolvo 2004).

The results of this systematic study are discussed in detail in the following subsections. Section 4.1 discusses the results related to the quality characteristics. It proposes a list of suitable quality characteristics for evaluating ubiquitous systems. Section 4.2 presents descriptive statistics about the software measures that have been proposed in the literature. The complete list of all measures found in this literature review and also quality characteristics related to some of these measures is presented in "Appendix." Section 4.3 discusses the results about the quality models. Section 4.4 presents the systematic map, which is basically two *x*–*y* scatter plots with bubbles in category intersections. The size of a bubble is proportional to the number of studies that are in a pair of categories corresponding to the bubble coordinates (Petersen et al. 2008). With this map, it is possible to see several research gaps in this area, which helps researchers in directing their efforts.

4.1 Quality characteristics proposed for ubiquitous systems' HCI evaluation

During the Data extraction process, we extracted: (1) characteristics present in the models and conceptual frameworks; (2) characteristics listed on papers even if they were not organized in a hierarchical way; and (3) issues described in the papers that are important to be considered in an HCI evaluation of a ubiquitous system but not explicitly presented by the papers' authors as a quality characteristic. We consider them all as characteristics in the Data extraction step. As previously presented (See Sect. 3.2.2.2), we identified only characteristics that impact on user interaction with ubiquitous systems. Figure 9 shows the number of quality characteristics presented in each study.

We found 134 quality characteristics from the first phase of the conducting activity, but with some limitations. Some papers propose just a list of characteristics without a clear definition of them (Kim et al. 2008; Sousa et al. 2011; Zhang et al. 2006). When they are defined, there is no consensus about them, and several studies use different names to the same goal, for instance: Kim et al. (2008) present the *Transparency* characteristic, but Scholtz and Consolvo (2004) call it *Invisibility* and Kourouthanassis et al. (2008) call it *Diffusion*. Another example, is the *Context-Awareness* characteristic: Some studies refer to it as *Context Sensitivity* (Ranganathan et al. 2005), *Contextualization Support* (Lee and Yun 2012; Lee et al. 2008), *Adaptability* (Kim et al. 2008) or even *Invisibility* (Scholtz and Consolvo 2004). Beyond that some characteristics had the same goal and different names, there are duplicated characteristics that need to be unified. For example, *Privacy* is cited by the following papers: Wu and Fu (2012), Abi-Char et al. (2010), Sun and Denko (2008),

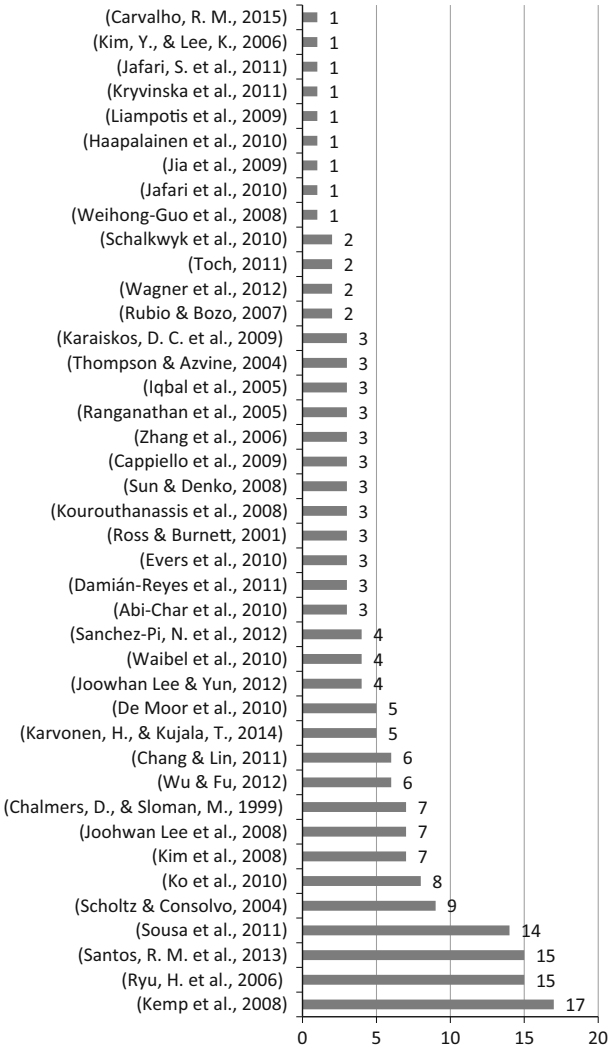


Fig. 9 Amount of extracted characteristics by paper

Scholtz and Consolvo (2004), Jafari et al. (2010), Toch (2011) and Liampotis et al. (2009). After the Peer Review 1, these problems were solved and we got 26 quality characteristics.

In the second phase of conducting activity, 52 quality characteristics were extracted. After Peer Review 2, among these 52 quality characteristics, we identified 33 characteristics identical to 26 existing characteristics (e.g., *Context-awareness*, *Mobility*, *Reliability*, *Privacy*), 3 characteristics as synonyms [e.g., *Unobtrusiveness*, *Perceived QoS* and *Connectivity* from Ryu et al. (2006)] and 4 characteristics as part of other characteristics [e.g., *Quality of Context*, *Natural Interaction Methods*, *Flexibility* and *Awareness Support* from Ryu et al. (2006)]. Furthermore, 11 characteristics were excluded. Four of them were related to HCI but not pertinent for ubiquitous systems because we consider the fact that ubiquitous systems should be transparent and calm and keep the user's attention on his/her

main activities. For example, *Controllability* is proposed by Ryu et al. (2006) and it explicitly requires user's perception and interaction with the system, which is not true when users are interacting with ubiquitous systems. Other 4 characteristics were excluded because they are closer to the measurement level than the characteristic level (e.g., *Bandwidth*, *Cost*, *Timeliness* and *Critically* from Chalmers and Sloman (1999), 1 characteristic also does not represent what really is a quality characteristic (*Support for everyday tasks* from Karvonen and Kujala 2014), 1 characteristic is very generic (*Accuracy* from Ryu et al. 2006), and 1 characteristic is related to the internal software quality (*Fault Tolerance* from Ryu et al. 2006). Only one characteristic was considered as a new one and was added to the set of characteristics (Reversibility from Ryu et al. 2006). Therefore, we got a final suitable set of 27 characteristics, which are presented in Table 5.

Usability, along with its subcharacteristics (*Efficiency*, *Efficacy* and *User satisfaction*), is the most referenced characteristic (13 papers). This information shows that *Usability* is the characteristic most commonly studied in HCI area (Ammar et al. 2015). The second most referenced is *Context-awareness* (12 papers), followed by *Transparency* (9 papers), *Privacy* (9 papers) and *Mobility* (7 papers).

Figure 9 presents the amount of extracted characteristics by each of 41 papers. The papers from which more quality characteristics were extracted are: Kemp et al. (2008), Ryu et al. (2006), Santos et al. (2013) and Sousa et al. (2011). Two of these papers are results from the first phase of our systematic mapping (Kemp et al. 2008; Sousa et al. 2011), and the other two are result from the backward (Ryu et al. 2006) and forward (Santos et al. 2013) snowballing.

Kemp et al. (2008) propose a set of heuristics to evaluate invisibility and usability in ubiquitous learning systems. Sousa et al. (2011) present a ubiquity measure that takes into account ubiquitous systems' technical capabilities, which we extracted as quality characteristics. Ryu et al. (2006) present characteristics to evaluate ubiquitous systems and middleware. They propose characteristics from the point of view of users and sensors. Finally, in Santos et al. (2013), we have presented our first results toward the definition of a quality model for HCI evaluation in ubiquitous systems.

Although these papers present most of the characteristics, many of them do not have any definition. It is important clearly define the quality characteristics in order to avoid misinterpretation. For example, *Analyzability*, *Interpretability* and *Credibility* from Sousa et al. (2011), and *Device Capability* and *Network Capability* from Santos et al. (2013) do not have definitions. Thus, it is not possible to understand clearly their meaning reading just the name of the characteristic.

We have noticed that several characteristics are generic for any kind of systems and they are already defined in the general standards of software product quality (ISO/IEC 25010 2011), for example Reliability, Safety, Trust, Availability, Effectiveness, Efficiency, User Satisfaction, Usability and Security. On the other hand, some characteristics are not presented in SQuaRE; thus, they are new ones, what makes us believe they are particular to the ubiquitous systems domain, such as Context-awareness, Mobility, Calmness, Transparency and Attention. However, since all characteristics found in our work are pertinent for ubiquitous systems evaluation and to keep answering the research questions with the same scope of the papers, we decided to keep the whole set of characteristics. An analysis of these characteristics to see which are generic or specific, by comparing to the standards, is presented in Sect. 5.

Table 5 Final set of quality characteristics

Characteristic	Definition	Synonyms/ similar characteristics	References
Acceptability	Represents the intention to use an application and its utilization rates	Adoption	Scholtz and Consolvo (2004)
		Potential customer acceptance	Ross and Burnett (2001)
		Acceptability	Waibel et al. (2010)
Attention	The ability to keep the user's attention to her/his main activity and not on the system and the technology involved	Attention	Wu and Fu (2012), Scholtz and Consolvo (2004)
		Distraction	Scholtz and Consolvo (2004)
		Focus	Kemp et al. (2008), Haapalainen et al. (2010), Santos et al. (2013)
		Unobtrusiveness	Ryu et al. (2006)
Availability	The service is always available, regardless of hardware, software or user fault, and it is often taken for granted until downtime occurs	Availability	Kryvinska et al. (2011), Ko et al. (2010)
Calmness	The ability to prevent users from feeling overwhelmed by information system	Calm technology	Wagner et al. (2012)
		Aesthetic and minimalist design	Kemp et al. (2008)
		Calmness	Santos et al. (2013), Carvalho et al. (2015)
Context-Awareness	The ability to perceive contextual information system and proactively adapt its functionality	Context-Awareness	Lee et al. (2008), Lee and Yun (2012), Toch (2011), Damián-Reyes et al. (2011), Ko et al. (2010), Kim and Lee (2006), Sanchez-pi and Carb (2012), Karaiskos et al. (2009), Santos et al. (2013), Karvonen and Kujala (2014)
		Contextualization	Lee et al. (2008), Lee and Yun (2012)
		Context sensitivity	Ranganathan et al. (2005)
		Adaptivity	Ko et al. (2010)
		Flexibility	Ryu et al. (2006)
Device capability	Properties of the device where the application will run (e.g., screen size, color depth, battery life)	Device capability	Zhang et al. (2006), Santos et al. (2013)
		Device	De Moor et al. (2010)
		Required HW/ SW	Wu and Fu (2012)
		Natural interaction	Karvonen and Kujala (2014)

Table 5 continued

Characteristic	Definition	Synonyms/ similar characteristics	References
Ease of use	The system should be easy to use by a target user group	Ease of use	Cappiello et al. (2009), Kemp et al. (2008), Santos et al. (2013)
		Facility of use	Rubio and Bozo (2007)
		Easy to use	Damián-Reyes et al. (2011)
		Perceived ease of use	Chang and Lin (2011)
Effectiveness	It refers to completeness in performing tasks	Effectiveness	Scholtz and Consolvo (2004), Kemp et al. (2008), Sanchez-pi and Carb (2012), Santos et al. (2013)
Efficiency	It refers to the amount of effort and resources required to reach a certain goal in the system	Performance	Cappiello et al. (2009), Schalkwyk et al. (2010)
		Efficiency	Scholtz and Consolvo (2004), Sanchez-pi and Carb (2012), Santos et al. (2013)
		Timeliness	Kemp et al. (2008)
		User's performance	Ranganathan et al. (2005)
Familiarity	User interactions with the system should improve the quality of her/his work. The user should be treated with respect. The design should be aesthetically pleasant	Familiarity	Iqbal et al. (2005), Santos et al. (2013)
Interconnectivity	An interconnected network between devices allows sharing	Interconnectivity	Kim et al. (2008), Karvonen and Kujala (2014)
Mobility	The ability to provide users with continuous access to resources and information system, regardless of their location within the limits of the systems	Mobility	Kim et al. (2008), Ryu et al. (2006)
		Ubiquity	Kourouthanassis et al. (2008), Karaiskos et al. (2009)
		Flexibility	Kemp et al. (2008)
		Multispace support	Ko et al. (2010)
		Connectivity and integrity	Ryu et al. (2006)
		Availability	Santos et al. (2013)
Network capability	Represents the collection of network information (e.g., signal strength, delay, jitter)	Network status	Zhang et al. (2006)
		Infrastructure	De Moor et al. (2010)
		Network	De Moor et al. (2010)

Table 5 continued

Characteristic	Definition	Synonyms/ similar characteristics	References
		Network capability	Santos et al. (2013)
		Perceived QoS	Chalmers and Sloman (1999)
Predictability	The ability, from past experiences, to predict the result of the system	Predictability	Kim et al. (2008)
Privacy	The ability to maintain information and data protected	Privacy	Wu and Fu (2012), Abi-Char et al. (2010), Sun and Denko (2008), Scholtz and Consolvo (2004), Jafari et al. (2010), Toch (2011), Liampotis et al. (2009), Jafari et al. (2011), Santos et al. (2013)
Reliability	The ability to maintain a particular level of performance when used under specific software conditions	Reliability	Waibel et al. (2010), Ryu et al. (2006), Chalmers and Sloman (1999)
Reversibility	The user's activities should be reversible to be able to restore to pre-existing states of the system.	Reversibility	Ryu et al. (2006)
Robustness	Degree to which a system or component can execute correctly in the presence of invalid inputs or stressful environmental conditions	Application robustness	Scholtz and Consolvo (2004)
Safety	The risk level of harming people, business, software, hardware, property or the environment in a specified context of use	Driver safety	Ross and Burnett (2001)
Scalability	The ability to provide services to a few or a large number of users	Scalability	Scholtz and Consolvo (2004)
Security	The protection to transport and to store information and also security controls who can access, use and modify context information	Security	Wu and Fu (2012), Abi-Char et al. (2010), Sun and Denko (2008), Ranganathan et al. (2005), Santos et al. (2013), Chalmers and Sloman (1999)
Simplicity	The user interface and the instructions should be simple	Simplicity	Kim et al. (2008), Ryu et al. (2006)
Transparency	The ability to hide the system, so users may not be aware of it. Moreover, the interaction is performed through natural interfaces	Diffusion	Kourouthanassis et al. (2008), Karaiskos et al. (2009)
		Invisibility	Scholtz and Consolvo (2004)
		Interaction transparency	Scholtz and Consolvo (2004)

Table 5 continued

Characteristic	Definition	Synonyms/ similar characteristics	References
		Invisibility	Kemp et al. (2008), Karvonen and Kujala (2014)
		Understandability	Thompson and Azvine (2004)
		Transparency	Ko et al. (2010), Ryu et al. (2006), Santos et al. (2013)
Trust	It is the belief of the user that the system uses your data properly and not cause any harm. It implies awareness, privacy and control	Trust	Sousa et al. (2011), Abi-Char et al. (2010), Jia et al. (2009), Sun and Denko (2008), Scholtz and Consolvo (2004), Evers et al. (2010), Santos et al. (2013)
		Trust/ethics/ responsibility	Kemp et al. (2008)
		Awareness support	Ryu et al. (2006)
Usability	The ability of the software to be understood, learned, used and attractive to the user, when used under specified conditions	Usability	Iqbal et al. (2005), Ross and Burnett (2001)
User satisfaction	The degree of user satisfaction and how the system is attractive for the user	User satisfaction	Sousa et al. (2011)
		Customer satisfaction	Cappiello et al. (2009)
		Appeal	Scholtz and Consolvo (2004)
		User satisfaction	Scholtz and Consolvo (2004)
		Satisfaction	Ranganathan et al. (2005), Sanchez-pi and Carb (2012), Santos et al. (2013)
Utility	The ability to provide value to user. The system provides a contribution to user that was not available before its development	Utility	Sousa et al. (2011), Chang and Lin (2011)
		Usefulness	Iqbal et al. (2005)
		Impact and side effects	Scholtz and Consolvo (2004)
		Perceived usefulness	Waibel et al. (2010)

4.2 Software measures proposed for ubiquitous systems' HCI evaluation

Regarding the identified measures, we should highlight that we did not perform a peer review, because most of measures were not clearly defined (in terms of measurements functions and/or quality measures elements) or even described in the selected papers, and were not applied in practice to make possible the understanding of their meaning.

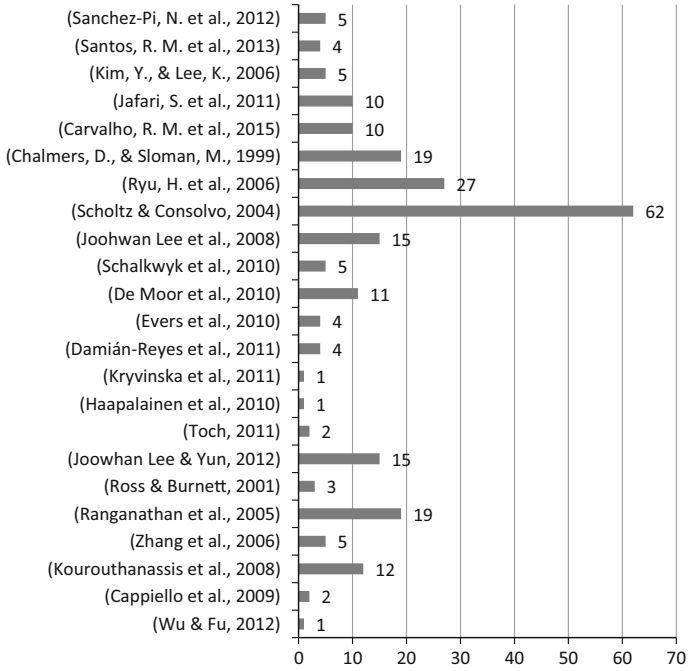


Fig. 10 Amount of extracted measures by paper

Therefore, we preferred just to organize them as a list of measures presented in "Appendix" as they were described in the papers. In this list, the measures were organized according to the characteristics they were defined to evaluate. For example, the "*Variety of supported contextual information*" measure was defined to evaluate the *Context-awareness* characteristic. However, the authors of some papers do not explicitly state the quality characteristic, i.e., there is no explanation about the characteristic that the measure aims to evaluate.

From the 41 studies, only 23 of them presented software measures. We extracted 218 measures from these studies. Figure 10 presents the number of measures proposed in each study.[3] The papers in which more software measures were extracted are Scholtz and Consolvo (2004), which proposes a framework of software measures, and Ryu et al. (2006), which provides measures to evaluate ubiquitous systems and middleware.

The main problems with these measures are related to the following: (1) Most of measures did not present any detail about how to compute them (measurement function or measure elements); (2) twenty-four measures the papers did not clarify to what characteristics they belong. These measures need to be better specified to make possible the understanding of their meaning (e.g., M126 "User participating degree for bidirectional communication of ubiquitous service while using service," see "Appendix"), (3) the measures are usually not validated, using, for example, case studies or controlled experiments, and (4) most measures are not documented using a more precise model such as the one defined by Fenton (Fenton and Pfleeger 1997) or by SQuaRE (ISO/IEC 25000 2014),

[3] The sum of the numbers in the graphic exceeds the total number of measures because there are duplicate measures among the papers.

which defined a structure to describe a measure that considers name, description, measurement functions and quality measures elements.[4] Using the concepts from SQuaRE (ISO/IEC 25000 2014), we classified the measures that we found in our work considering their defined measurements function, since it allows to identify the quality measurement elements, as follows:

A. *Well defined* We found 39 measures (see "Appendix") that have a measurement function and more than one quality measures elements. For example, Automation: (A/A + B) where A is the number of decisions taken autonomically by the decision engine and B is the number of decisions sent to the user (Toch 2011).
B. *Defined but without measurement function* 64 measures are defined in terms of just a property to quantify, e.g., ratio of error occurrence during service use (Lee et al. 2008).
C. *Not defined* 115 measures do not have any information about, (e.g., degree of adaptation to contextual changes (Kourouthanassis et al. 2008) being difficult to infer what they are really measuring.

Regarding the quality characteristics they are intended to evaluate, most measures found are defined to evaluate the Context-awareness characteristic (44 measures). We believe that an explanation for this is that context-awareness is an indispensable characteristic for a ubiquitous system. In general, these measures are concerned with the correctness of the collected context, e.g., probability that an instance of context accurately represents the corresponding real-world situation, defined by Damián-Reyes et al. (2011), and with the benefits that context-awareness brings to the user, e.g., reduction in the number of configuration actions that the user has to take to configure an environment in a context-sensitive manner, defined by Ranganathan et al. (2005).

The second characteristic most cited by measures was *Usability* (24 measures). Eight measures of *Usability* are specific to evaluate the subcharacteristic *User Satisfaction* and four of them to evaluate *Efficiency*. The remaining of them (12 measures) is not defined for any subcharacteristic of usability.

The other characteristics most cited were *Network Capability* (19 measures), *Transparency* (15 measures), *Acceptability* (15 measures), followed by the *Attention* characteristic (12 measures), *Privacy* (12 measures), *Calmness* (10 measures) and *Trust* (10 measures), which indicates that these characteristics are relevant for measurement in ubiquitous systems' HCI evaluations.

There are some measures that can evaluate more than one characteristic. For example, the M119 (*User control over private information: content privacy*, *identity privacy and location privacy*) and M120 *Expressiveness of the security policy: Support for mandatory and discretionary rules*, *context sensitivity*, *uncertainty handling*, *conflict resolution*) measures evaluate both *Privacy* and *Security* characteristics.

[4] ISO/IEC 25000 2014 presents the following definitions:
Quality measure element (QME): measure defined in terms of a property and the measurement method for quantifying it.
Property to quantify: property of a target entity that is related to a quality measure element and which can be quantified by a measurement method
Quality measure (QM): derived measure that is defined as a measurement function of two or more values of quality measure elements.
Measurement function: algorithm or calculation performed to combine two or more quality measure elements.

4.3 Quality models for ubiquitous systems' HCI evaluation

To answer the ***RQ3***, we analyzed whether the paper had measures, subcharacteristics and characteristics organized in a hierarchical way, like in the SQuaRE standard (ISO/IEC 25010 2011). It could be an empirical model or a framework. As mentioned previously, from a number of 41 papers, only 3 had described a solution, like conceptual frameworks and models of characteristics: Scholtz and Consolvo (2004), Lee et al. (2008) and Santos et al. (2013). Although two of these studies (Scholtz and Consolvo 2004; Lee et al. 2008) do not call explicitly such solutions as quality models, they use them to make quality evaluations, so we considered them as quality models.

The first study (Scholtz and Consolvo 2004) aims to develop a user evaluation framework of ubiquitous applications. It defines nine evaluation areas (*Attention*, *Adoption*, *Trust*, *Conceptual Models*, *Interaction*, *Invisibility*, *Impact and Side Effects*, *Appeal* and *Application Robustness*), which were divided into metrics, which in turn were divided into conceptual measures. For example, in the *Attention* area, two metrics were created, Focus and Overhead. For the Overhead metric, two measures were created: *percent of time user spends switching among foci* and *workload imposed on user attributable to focus.* The main limitation of this study is the poor definition of their measures. They have only names; properties as measurement function and values of interpretation are not defined. For example, how does one calculate exactly the *percent of time user spends switching among foci?* How does one interpret the results of this measure?

The second study (Lee et al. 2008) aims at developing user-centered evaluation metrics to evaluate ubiquitous service interactivity attributes. The attributes defined were: *Contextualization support*, *Service capability*, *Ubiquity support and User experience support*. Those attributes were divided into 15 measures. However, this model has some limitations. First, it does not have a good hierarchy definition: Some attributes comprise a lot of information (*e.g.*, service capability means performance, security and storage abilities), so it should be decomposed into subfactors. Second, it does not define all characteristics necessary to do HCI evaluations of ubiquitous systems. For example, they do not mention anything about trust, which is a very important characteristic that has a relevant impact on the user interaction. If the user does not trust in the system, hardly will he/she use it for his/her daily activities as defended by Abi-Char et al. (2010), Evers et al. (2010), Jia et al. (2009), Scholtz and Consolvo (2004), Sousa et al. (2011) and Sun and Denko (2008). Another example concerns the characteristics related to resource limitation, like hardware resource, and transparency interaction (defended by Scholtz and Consolvo 2004; Wu and Fu 2012).

In the third study (Santos et al. 2013), we have proposed a quality model to HCI evaluation in ubiquitous systems. This model consists of characteristics and subcharacteristics that have impacts on user interaction quality and measures capable of evaluating them for a particular system. This model has characteristics specific for ubiquitous systems interaction (context-awareness, transparency, attention, calmness and mobility). However, this paper presents only measures for context-awareness.

We noted that the existing quality models are not complete. None of them defines completely the measures. Two of them (Scholtz and Consolvo 2004; Lee et al. 2008) do not describe how we can collect the measures and how we can interpret them, following the SQuaRE standard (ISO/IEC 25000 2014). Besides this, they fail to define some issues specific to HCI evaluation of ubiquitous systems, as discussed previously.

4.4 The systematic map

In order to create our systematic map, we classified all papers according to the defined facets: the contribution type facet (see Table 2) and the research type facet (see Table 3). Figures 11 and 12 present the results of these classifications. In the last step of SM, we crossed the data from these two classification facets and we generated the systematic map presented in Fig. 10. According to Petersen et al. (2008), this analysis enables presenting the frequencies of studies for each category.

Analyzing the systematic map presented in Fig. 10, we concluded that:

- Most of the papers (71 %) are "Solution Proposal." No study was classified as "Experience Papers." Only six papers (15 %) were classified as "Validation Research," i.e., there are experiments to validate the proposed work, and, finally, only one (2 %) is "Evaluation Research," what means the proposition is widely used in the industry;
- The only paper classified as "Evaluation Research" defines software measures to evaluate the Google Search by Voice, a company and system widely known, which means this work was implemented in practice;
- The most of the "Validation Research" papers are classified as "Measures Framework" because they use their measures in experimental studies, which means they have not yet been implemented in practice but only in experiments;
- Most of the "Solution Proposal" papers are classified as "Measures Framework" (13 papers—45 %), but we also found papers classified as "Quality Issues" (7 papers—24 %), "Characteristic Framework" (7 papers—24 %) and "Quality Model" (2 paper—7 %). However, most of these studies that contain measures do not detail how to measure the ubiquitous system. Their software measures do not have measurement functions, interpretation values and collection methods. The papers that propose characteristics or quality issues do not present a pattern in their definitions; some of them do not provide a clear definition and others do not take into account important specific characteristics of ubiquitous systems like context-awareness; and
- Only three papers are classified as "Quality Model," which we considered the most complete contribution since it contains characteristics, subcharacteristics and measures. However, from these papers, two are classified as "Solution Proposal" and the other as "Philosophical Papers." Then, there is no quality model for ubiquitous systems that has been validated and/or used in practice.

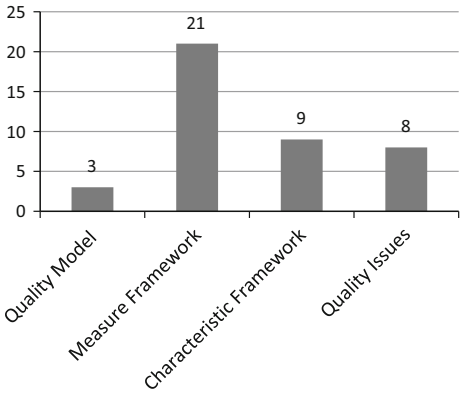


Fig. 11 Distribution of selected papers structured by contribution of the paper

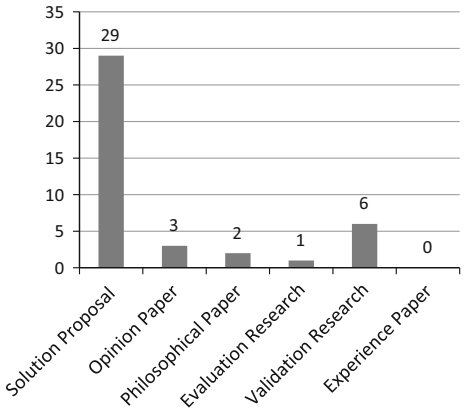


Fig. 12 Distribution of selected papers structured by research type

Based on this analysis, we can say that the HCI measurement area in ubiquitous systems still has many gaps. Therefore, more research is needed in the following aspects:

- Aggregating all characteristics found in these studies of the SM. They could be complementary, because they do not present all important ubiquitous features (e.g., Scholtz and Consolvo 2004 does not consider availability, and Lee et al. (2008) do not consider transparency). Thus, some of them have important characteristics that the others do not have;
- Although a lot of measures have been proposed, they need to be better specified, which means a clear and complete definition of their function, procedure collection, interpretation collection and so on. These measures must be defined using a format of the documentation from the SQuaRE standard (ISO/IEC 25010 2011). This format ensures that the measure provides all the necessary information to an evaluator to collect it;
- A complete quality model that organizes characteristics and measures in a hierarchical way should be proposed and evaluated. The characteristics and measures found in this paper can be a start to define it. We proposed a first model in this direction (Santos et al. 2013) organized into four characteristics (*Trustability*, *Resource-limitedness*, *Usability and Ubiquity*), and the other ones were included as subcharacteristics. However, this model was only a first proposition and we are still working on it to define consistent measures for all quality characteristics that could really applied for different ubiquitous systems is a long-term research; and
- Validating software measures using, for example, case studies or controlled experiments. As mentioned by Montagud et al. (2012), a validation can corroborate that the measure is measuring what it expects to measure.

5 Discussion

All characteristics found in the SM impact, somehow, on the quality of interaction with ubiquitous systems and, therefore, need to be evaluated. In general, we can say that all characteristics are defined taking into account that the interaction with ubiquitous systems needs not only to be efficient and effective, but also transparent and implicit to the user.

One can say that some of the quality characteristics do not appear to be oriented to the user interaction, but we defend that they affect the interaction. For example, *Scalability* can be only perceived by system administrators and maintainers. However, in a ubiquitous environment, a ubiquitous system can be used by many users and by other systems everywhere and any time, low scalability has a high user-perceivable impact.

To better analyze these characteristics, we consider the SQuaRE standard (ISO/IEC 25010 2011)[5] that organizes quality characteristics in two quality models:

- System/Software Product Quality Model (left side of Fig. 11)—composed of eight characteristics, which are further subdivided into subcharacteristics that can be measured internally or externally.
- Quality in Use Model (right side of Fig. 11)—composed of five characteristics which are further subdivided into subcharacteristics that can be measured when a product is used in a specific context. These characteristics are related to the outcome of an interaction when a product is used in a particular context of use.

We therefore analyze the 27 characteristics presented in the previous section in comparison with these models. First, we identified that several characteristics found in the SM are already defined with the same name in the System/Software Product Quality Model, as follows: *Usability*, *Reliability*, *Availability and Security*. These characteristics have the same definitions; however, there are particularities about ubiquitous systems that have to be taken into account for evaluating their quality. For example, *Security* in general means the degree to which a product or system protects data and information, but in ubiquitous systems, security has also to handle with important data (e.g., location, profile) being collected all the time by other systems. Thus, software measures for *security* in ubiquitous systems have to be defined considering these particularities.

Moreover, regarding the System/Software Product Quality Model, we found characteristics that can be mapped as synonyms to existent characteristics from SQuaRE. *Privacy*, for example, can be mapped to the *Confidentiality* subcharacteristic. Some characteristics can be included in others, for example: *Device Capability* and *Network Capability* of the *Capacity* subcharacteristic, *Robustness* as subcharacteristic of *Reliability* and *Interconnectivity* as subcharacteristic of *Compatibility*.

Considering the Quality in Use Model, we found the following characteristics defined with the same name and meaning: *Effectiveness*, *Efficiency*, *Satisfaction* and *Trust*. We also considered that the characteristic *Safety* could be mapped as synonym to *Freedom from Risk*. Some characteristics such as *Familiarity*, *Reversibility* and *Simplicity* can be subcharacteristics of *Usability*.

Also, one can say that the C*ontext-awareness* characteristic is similar to *Context coverage* characteristic from the Quality in Use Model. Nevertheless, these characteristics have different meanings. *Context coverage* is the degree to which a product can be used efficiently, effectively, with freedom from risks and with satisfaction in specified contexts of use and other contexts not initially identified. Moreover, *Context-awareness* means the system's capacity to collect context information and to use it to make dynamic and/or static adaptations.

[5] We could use the standard ISO/IEC 9241, specific for HCI, for our analysis. However, we chose to work with the SQuaRE since it aggregates the other quality standards (ISO/IEC 9126 2001) and (ISO/IEC 14598 1999), and the characteristics from ISO/IEC 9241 are also all defined in it.

Other characteristics that are presented in Nielsen (1994) for user interface evaluation (*Acceptability*, *Utility*, *Usability* and *Ease of Use*) are usually used to evaluate the Quality in Use Model characteristics.

With this analysis, we concluded that twenty-one characteristics are generic for any kind of system, not only for ubiquitous systems. However, they should be particularized for a ubiquitous system when applied in an evaluation. On the other hand, six characteristics are not presented in SQuaRE and, thus, they are new ones, which make them particular to the ubiquitous systems domain, and they are: *Context-awareness*, *Mobility*, *Calmness*, *Transparency*, *Attention and Predictability*. This suggests that one should give more attention to these characteristics when evaluating the quality of interaction with ubiquitous systems.

6 Threats to validity

Although we used a systematic and rigorous process of literature review, there are some threats to the validity (limitations) of the results of our study. In the next subsection, we discuss the relevant threats to our study according to the four categories of validity threats for software engineering research proposed by Petersen and Gencel (2013): (1) descriptive validity, which is the extent to which observations are described accurately and objectively; (2) theoretical validity, which is determined by our ability of being able to capture what we intend to capture; (3) generalizability, related to the internal generalizability (within groups, communities or a company) and external generalizability (between groups or organizations); and (4) interpretive validity, which is achieved when the conclusions drawn are reasonable given the data. This classification is recommended by the guideline for systematic mapping presented from Petersen et al. (2015), and it helps to consistently report the threats.

In the next subsections we discuss relevant threats to validity of our systematic mapping results according to these categories.

6.1 Descriptive validity

In this category, we have the threats related to factual accuracy of the account. A possible threat to this validity could be an inaccuracy of Data extraction, which may induce to wrong and/or incomplete results. For example, a quality characteristic may not have been identified during the Data extraction or a characteristic that is not well defined may have been extracted. To mitigate this threat, a Data extraction form (See Table 4) has been designed based on our research questions to support the recording of data. Moreover, this form contains definitions and examples about what is a quality characteristic, a software measure, a quality model and an application domain, which makes the Data extraction process clear and objective. The reviewers extracted the data (name and definition of the quality characteristic, software measures and application domain) exactly how presented in the primary study. Also, a peer review (see Sect. 3.2.1.4) was performed to obtain a final suitable set of characteristics, and therefore, all characteristics were analyzed by three researchers. Another threat could be the bias in the identification of the research and contribution type. To mitigate this threat, a glossary and examples were discussed among the researchers, and in case of doubts, the classification was also performed by other researcher.

6.2 Theoretical validity

The threats in this category are concerned with whether we captured what we intend to capture (Petersen et al. 2015). To reduce these threats, we designed a systematic mapping protocol including all important activities proposed by guidelines for systematic studies, such as, a search string based on our research questions, inclusion/exclusion criteria and a Data extraction form.

The missing of studies due to a not well-designed search string is one relevant threat in this category, since it implies in incomplete results. To reduce this threat, we defined our search string using well-established terms in the community (e.g., measures) as well as synonymous and alternative words (e.g., metrics, see Table 1). We also performed several tests with the search string to assess whether the resulting papers answer our research questions. Other reason to the missing of studies could be that we did not use other terms from concepts related to ubiquitous systems, like "mobile" and "context aware," which implies the lack of important papers. However, the use of such keywords could bias our focus that is ubiquitous systems. Furthermore, we believe that the lack of such keywords did not harm our findings since we had in our findings, for example, several papers related to mobile applications that are ubiquitous and, based on these papers, we identified some relevant characteristics, for example: mobility, device capability and context-awareness. With a first set of papers that answered our search question, we performed also snowballing procedures (backward and forward) aiming to find out as many as possible candidates papers for our mapping study.

Other threat is that we did not conduct manual searches. This threat could imply in not identify results of research that propose new characteristics and measures for HCI in ubiquitous systems. To reduce this threat, we have searched broadly in six well-known online databases (ACM Digital Library, IEEE Xplore, Scopus, Science Direct, SpringerLink and Compendex) that index the most well-reputed publication events in software engineering, ubiquitous computing and human–computer interaction. Besides, we also applied backward and forward snowballing from the list of each primary study aiming avoid the loss of some important paper. For the forward snowballing, we used a broader indexer of papers (Google Scholar) that covers not only papers indexed by journals and conferences, but also by internal periodicals and technical reports from the research institutions.

We also could have researcher biases during the selection and extraction of data. We mitigate these threats by using a rigorous study selection process, as described in the research protocol, and an extraction form. Besides, both selection and extraction process were performing by peers (one researcher reviewed the selection/extraction of the other) in consensual meetings. When there was a disagreement between the reviewers, a third reviewer helped to reach a consensus.

We also highlight that no restriction was defined for the kind of study (e.g., book and short paper) considered in our mapping. Although we do that aiming to make a broad overview of the area, we cannot ensure that all relevant literature (e.g., books) has been included, which implicates that our set of characteristics, measures and models could be incomplete and our general conclusions based on the systematic map could be not accurate. For instance, if a study is not indexed by the databases used or did not contain the key terms, it will not be identified by our search string.

Moreover, as we conducted the searches in the online databases in the middle of 2013, we could have missed some interesting paper published after 2012. To reduce this threat,

we performed a forward snowballing (until 2015) from the 32 papers obtained by the first phase of our conducting activity (see Fig. 2). This does not eliminate the threat of existing an interesting paper to our mapping published after 2012 that was not captured by our databases searchers, but we believe that the set of papers resulting from the online searches and snowballing is representative.

However, by performing snowballing (backward and forward), we observed that only one new characteristic was included in the initial set of quality characteristics identified by the database search. All papers from forward snowballing (that got papers till end of 2015) addressed quality characteristics similar to the 26 ones previous defined. This fact may suggest that we obtained a representative set of quality characteristics, and that maybe new papers would converge in the same set.

At last, we performed the second phase of conducting activity almost 3 years after we conducted the first phase. This could imply in inconsistent results if the Data extraction were performed in different way at both phases. To mitigate this threat, we used the same Data extraction form and we performed a peer review with the same researchers at both conducting activities to ensure consistent results. Although we recognize this treat, we argue that the snowballing could suggest that our set of quality characteristic is well representative about HCI quality in ubiquitous systems.

6.3 Generalizability validity

The validity threats in this category are concerned with the ability to generalize the results of the systematic mapping. Petersen and Gencel (2013) present two generalizability types: external generalizability (between groups, organizations, different populations) and internal generalizability (within a group, a company). To mitigate the threats related to internal generalizability, we used a systematic mapping process, which is a rigorous process of literature review. As a result, a wide range of measures described in 41 papers published between 1999 and 2015 have been identified. Thus, we believe that the internal generalizability is not a major threat. Regarding the external generalizability, we may not guarantee the applicability of the measures identified to any kind of applications (e.g., mobile application, internet of things system). The identified measures were, in some papers, applied in specific kinds of systems. However, our study is focused on the ubiquitous computing and, therefore, we cannot generalize our results for other domains.

6.4 Interpretive validity

The threats in this category are concern with whether the conclusions were based on the data. A threat in this category is the researcher bias. To reduce the subjective interpretations of the researchers, the Data extraction and the data analyze were performed by two reviewers (or three in the case of the two reviewers disagree in something) for each included paper.

In regard to the classification of the research type, we followed the scheme from Wieringa et al. (2005) to classify the research type of the included papers in six categories as done in many other secondary studies in software engineering. However, since this classification is defined on a higher abstraction level (Wohlin et al. 2013), in some of the included papers, other researchers could disagree of our classification and classify them in other category. In addition, other researchers may possibly come up with different classification schemes, but we believe that the scheme used in this paper was enough to answer our research questions. To mitigate this bias, we also performed this classification by peers.

Moreover, we also considered another classification (contribution type facet), defined by us. This scheme was defined by keywording relevant topics in the abstract of the papers as described in the guidelines of Petersen et al. (2008). We believe that both schemes used in this paper (the classification of Wieringa and our classification) were suitable to answer our research questions.

Finally, other threat is related to biases in the peer review process to analyze the characteristics (see Sects. 3.2.1.4 and 3.2.2.3), which has as goal the identification of synonyms and the definition of a list of different quality characteristics. To mitigate such threat, this process was conducted by three researchers during several meetings. All of them are expert in the quality and ubiquitous computing area. One researcher is a PhD student who works on the software quality and ubiquitous computing areas. Two others are professors: One has more than 15 years of experience in definition and use of software measures, and the other has more than 10 years of experience in mobile and ubiquitous computing.

7 Related work

To the best of our knowledge, this is the first review on quality characteristics and measures for the HCI evaluation of ubiquitous systems. However, there are several papers that present systematic reviews or systematic mapping of literature in different domains and with different purposes, for instance, software product line testing (Silveira et al. 2011), software evolution visualization (Novais et al. 2013), security in cloud computing (Da Silva et al. 2013). Considering the investigated topics in our study (ubiquitous applications, quality characteristics/quality models and HCI), we found four relevant studies and we presented them in this section.

The first one is from Spínola and Travassos (2012) and concerns about ubiquitous systems. They present a conceptual framework to support the characterization of ubiquitous software projects according to their ubiquity adherence level. This work follows a research strategy based on systematic reviews and surveys to acquire UbiComp knowledge and organize a conceptual framework. They identified 11 UbiComp features, for example: context sensitivity, adaptable behavior, service omnipresence, heterogeneity of devices, experience capture, spontaneous interoperability, scalability, privacy and trust, fault tolerance, quality of service and universal usability. Although this paper considers the same domain as ours, which is ubiquitous computing, it differs from our review since the focus of the identified characteristics is not on the HCI aspect of ubiquitous systems and does not identify measures for evaluation of ubiquitous systems.

In the second one, Montagud et al. (2012) had the goal to identify quality characteristics (or attributes) and measures, but not for the same domain as ours (ubiquitous systems), and they focused on the software product line domain. The attributes and measures were classified using a set of criteria that includes the phase of the life cycle in which the measures are applied. At the end of this study, a catalog was elaborated identifying quality attributes and measures for all development phases of SPL and the final product.

The third one, Oriol et al. (2014), is also related to quality characteristics, more specifically quality models. They evaluated the current state of the art of the existing quality models for web services. In total, they found 47 different quality models. One of their results is that most of the quality models do not take into account standards like ISO/IEC 25000 for the development of the proposed quality model. We concluded the same in

our review. This work differs from ours because they evaluated a different domain (web services). Moreover, they do not identify software quality measures.

Recently, Reis (2015) performed a systematic mapping looking for approaches to evaluate usability for mobile applications. They identified 101 usability evaluation approaches for mobile applications, 28 of which applied to ubiquitous mobile applications. One of the goals of this work was to verify the presence of ubiquity characteristics, defined by Spínola and Travassos (2012), in the approaches found in their work. They figured out that 5 from 11 ubiquity characteristics were considered in the approaches that they found. Moreover, they also look for what the approaches considered in terms of usability (that is, memorability, cognitive load, errors, learnability, efficiency, effectiveness and satisfaction). Although this paper looks like ours while considering aspects about usability and ubiquitous application, it differs from ours in the focus of identifying approaches for evaluation and not quality characteristics and measures and it was particular for mobile applications not ubiquitous applications that have specific characteristics discussed in our work (Figs. 13, 14).

8 Conclusion and Future work

Characteristics of ubiquitous systems such as context-awareness and invisibility bring new challenges to human–computer interaction. In this scenario, the following question arises: *How can we evaluate the HCI quality in ubiquitous systems?*

Motivated by this question and by the scarcity of work in the literature that summarizes all evidence about HCI evaluation in ubiquitous systems, we decided to conduct a systematic mapping study in the context of HCI quality evaluation in ubiquitous systems. The search was conducted in six important electronic databases, and after following a rigorous

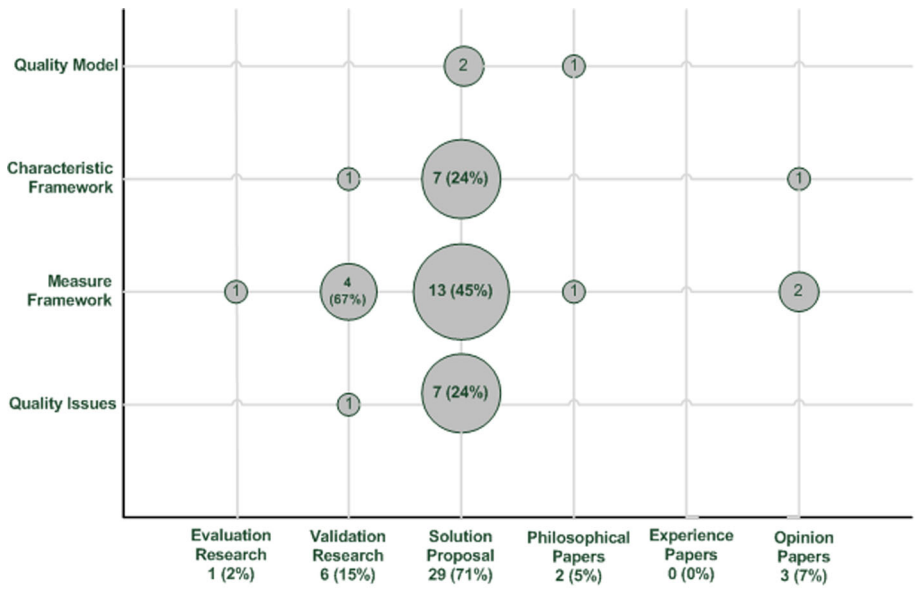


Fig. 13 The systematic map

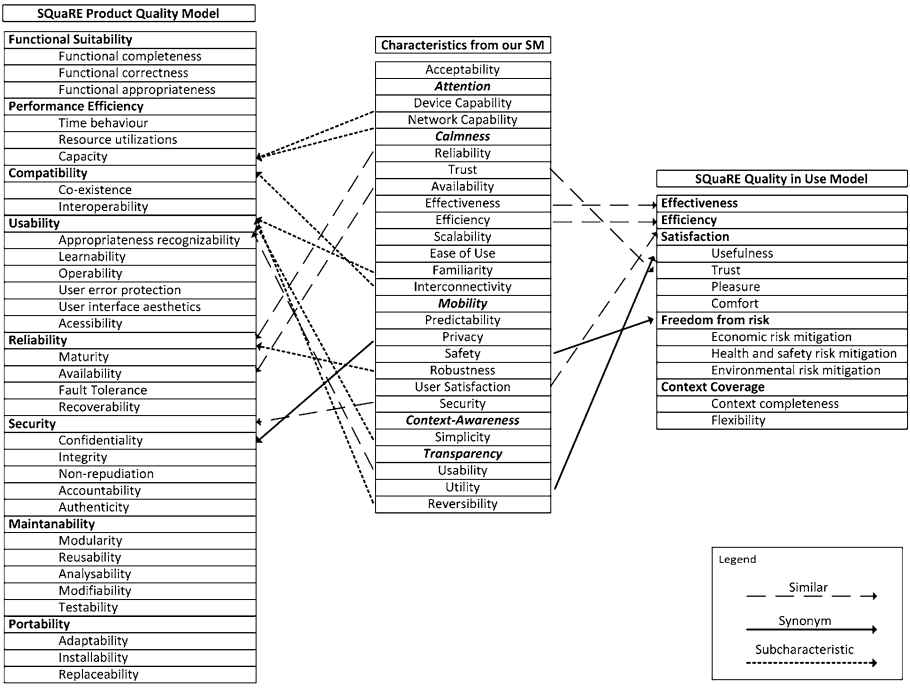


Fig. 14 System/Software Product Quality Model (ISO/IEC 25010 2011)

literature review process, we produced a set of 27 quality characteristics and 218 measures useful for HCI quality evaluation in ubiquitous systems.

We then analyzed these 27 characteristics considering the System/Software Product Quality Model and Quality in Use Model from the SQuaRE standard. We concluded that 21 characteristics are generic for any kind of system, since they are present in the SQuaRE models, but it is still necessary to take into account ubiquity particularities when evaluating these characteristics in ubiquitous systems. On the other hand, we identified six characteristics that make us believe they are particular to HCI evaluation of ubiquitous systems.

The SM results also show evidence about the need of more research in order to: (1) aggregate all characteristics found in these SM studies; (2) specify better the identified measures; (3) validate the identified measures; and (4) create a complete quality model that organizes characteristics and measures in a hierarchical way.

Furthermore, this study leaves the following future work:

- The measures may have relationships, for example, measures from Context-awareness may impact on Usability.
- Once new measures are defined, they need to be applied in different domains and different types of ubiquitous systems (e.g., touristic guides) to be really validated and to show their utility;
- Knowing the quality characteristics, new approaches for the development of ubiquitous systems that consider the quality characteristics early in the development should be defined instead of leaving the quality evaluation for the end of the development cycle;

- Prioritization of the software measures can be done with respect to their importance;
- Software testing procedures should be defined to correctly collect each one of the measures in different scenarios of use; and
- Definition of a testing process for different types of ubiquitous systems based on both the quality characteristics and the software measures presented in this paper.

Acknowledgments We thank FUNCAP (Ceará State Foundation for Support of Scientific and Technological Development, Brazil) and CNRS (Centre National de la Recherche Scientifique, France) for the financial support of this work, which is a result of the Maximum Project—A Measurement-based Approach for the Quality Evaluation of Human–Computer Interaction in Ubiquitous Systems, under grant number INC-0064-00012.01.00/12. We also thank CAPES for sponsoring Rainara Maia Carvalho and Ismayle de Sousa Santos with PhD scholarships, and CNPq for sponsoring Rossana Maria de Castro Andrade with a Researcher Scholarship - DT Level 2.

Appendix: Software Measures

This appendix presents the 219 extracted software measures from the systematic mapping presented in this paper. These measures are classified and organized in tables according to the quality characteristics that they are aimed at evaluating (Tables 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27).

Table 6 Software measures for acceptability

ID	Software measure	Classification	References
M1	Rate: new users/unit of time	Well defined	Scholtz and Consolvo (2004)
M2	User rationale for using the application over an alternative	Not defined	Scholtz and Consolvo (2004)
M3	Technology usage statistics	Not defined	Scholtz and Consolvo (2004)
M4	Changes in productivity	Not defined	Scholtz and Consolvo (2004)
M5	Perceived cost/benefit	Not defined	Scholtz and Consolvo (2004)
M6	Continuity for user	Not defined	Scholtz and Consolvo (2004)
M7	Amount of user sacrifice	Not defined	Scholtz and Consolvo (2004)
M8	User willingness to purchase technology	Not defined	Scholtz and Consolvo (2004)
M9	Typical time spent setting up and maintaining technology	Defined but without measurement function	Scholtz and Consolvo (2004)
M10	Number of actual users from each target user group	Defined but without measurement function	Scholtz and Consolvo (2004)
M11	Technology supply source	Not defined	Scholtz and Consolvo (2004)
M12	Categories of users in post-deployment	Not defined	Scholtz and Consolvo (2004)
M13	Number of tasks user can accomplish that were not originally envisioned	Defined but without measurement function	Scholtz and Consolvo (2004)

Table 6 continued

ID	Software measure	Classification	References
M14	User ability to modify as improvements and features are added	Not defined	Scholtz and Consolvo (2004)
M15	Intent to use the system	Not defined	Evers et al. (2010)

Table 7 Software measures for attention

ID	Software measure	Classification	References
M16	The interaction level of a space is defined as the degree of foreground interaction between a user and this space recently	Not defined	Wu and Fu (2012)
M17	Number of times a user must change focus due to technology	Defined but without measurement function	Scholtz and Consolvo (2004)
M18	Number of displays/actions users need to accomplish, or to check progress, of an interaction	Defined but without measurement function	Scholtz and Consolvo (2004)
M19	Number of events not noticed by a user in acceptable times	Defined but without measurement function	Scholtz and Consolvo (2004)
M20	Percent of time user spends switching among foci	Defined but without measurement function	Scholtz and Consolvo (2004)
M21	Workload imposed on user attributable to focus	Not defined	Scholtz and Consolvo (2004)
M22	Distraction: time taken from the primary task; degradation of performance in primary task; level of user frustration	Defined but without measurement function	Scholtz and Consolvo (2004)
M23	Degradation of performance in primary task	Not defined	Scholtz and Consolvo (2004)
M24	Level of user frustration	Not defined	Scholtz and Consolvo (2004)
M25	Significant visual distraction (workload)	Not defined	Ross and Burnett (2001)
M26	Number of attention switches	Defined but without measurement function	Haapalainen et al. (2010)
M27	A/UOT where A = number of times that the user staring at the input or output artifacts to trigger their next activities, UOT = user operating time during observation period	Well defined	Ryu et al. (2006)

Table 8 Software measures for availability

ID	Software measure	Classification	Reference
M28	Availability = mean time between failures/mean time between failures + mean time to repair	Well defined	Kryvinska et al. (2011)

Table 9 Software measures for calmness

ID	Software measure	Classification	Reference
M29	Adaptation degree $X = \frac{\left(\sum_{j=1}^{N} \frac{Aj}{Bj}\right) * 100}{N}$ N = Number of the different adaptations Aj = Number of times the system adapts Bj = Number of times an adaptation j was requested (the context changed) Closer to 100 % is better	Well defined	Carvalho et al. (2015)
M30	Adaptation correctness degree $X = \frac{\left(\sum_{i=1}^{N} \frac{Ai}{Bi}\right) * 100}{N}$ Ai = Number of correctly performed adaptations i Bi = Number of performed adaptations i N = Number of the different adaptations The closer to 100 % is better	Well defined	Carvalho et al. (2015)
M31	Indicator of Transparent Mobility X = A, where A is (0) Nonexistent, (1) Low, (2) Medium, (3) High	Well defined	Carvalho et al. (2015)
M32	Availability degree X = B, where B is the mode of (1) High, (2) Medium, (3) Low, (4) Very Low	Well defined	Carvalho et al. (2015)
M33	Context-awareness timing degree X = C, where C is the mode of (0) Nonexistent, (1) Low, (2) Medium, (3) High	Well defined	Carvalho et al. (2015)
M34	Number of irrelevant focus changes = X = A, where A = number of actions that changes user's focus during use of the application The further away from 0 is better	Well defined	Carvalho et al. (2015)
M35	Proactivity of the application X = N − A where N = Number of total actions developed that can be supported by sensors A = Number of actions the application replaces	Well defined	Carvalho et al. (2015)
M36	Number of Failures = X = N, where N = Total number of failures have occurred	Well defined	Carvalho et al. (2015)
M37	Relevancy Degree X = B, where B is the mode of (1) High, (2) Medium, (3) Low, (4) Very Low	Well defined	Carvalho et al. (2015)
M38	Courtesy Degree X = D, where D is the mode of (1) High, (2) Medium, (3) Low, (4) Very Low	Well defined	Carvalho et al. (2015)

Table 10 Software measures for context-awareness

ID	Software measure	Classification	References
Part 1			
M39	Variety of supported contextual information	Not defined	Kourouthanassis et al. (2008)
M40	Degree of adaptation to contextual changes	Not defined	Kourouthanassis et al. (2008)
M41	Degree of proactive system operation	Not defined	Kourouthanassis et al. (2008)
M42	Number and diversity of access devices and sensor technologies	Defined but without measurement function	Kourouthanassis et al. (2008)
M43	Degree of integration and coordination among device types and IT artifacts	Not defined	Kourouthanassis et al. (2008)
M44	The degree of receiving unexpected service by providing unrequested service functions: whether the implicit needs of ubiquitous service user is to be provided or not	Not defined	Lee et al. (2008)
M45	Recognized degree of wasting time during the use of ubiquitous service	Not defined	Lee et al. (2008)
M46	Ratio to change the service contents properly to user's preference or habit automatically	Not defined	Lee et al. (2008)
M47	Required time/degree for user to modify the service function procedure in the user's convenience	Not defined	Lee et al. (2008)
M48	Distance degree that user must move additionally to receive the service at proper place and time	Not defined	Lee et al. (2008)
M49	Range degree of physical spaces to recognize the status of the service	Not defined	Lee et al. (2008)
M50	Ratio of error occurrence during use of the service	Defined but without measurement function	Lee et al. (2008)
M51	Confidence: expressed as a probability that the context has been sensed or deduced correctly	Defined but without measurement function	Ranganathan et al. (2005)
M52	Accuracy: expressed as an error percentage of the sensed or inferred contexts	Defined but without measurement function	Ranganathan et al. (2005)
M53	Freshness: measured as the average time between readings of a certain kind of context	Defined but without measurement function	Ranganathan et al. (2005)
M54	Resolution: the area within which location information can be narrowed down to room-level, building-level, for example	Not defined	Ranganathan et al. (2005), Sanchez-pi and Carb (2012)

Table 10 continued

ID	Software measure	Classification	References
M55	Reduction in number of configuration actions that user has to take to configure environment in a context-sensitive manner	Defined but without measurement function	Ranganathan et al. (2005)
Part II			
M56	Reduction in number of times user was disturbed or annoyed by a proactive action taken by the system or by a notification. This is measured based on user feedback	Defined but without measurement function	Ranganathan et al. (2005)
M57	Reduction in number of configuration actions	Defined but without measurement function	Ranganathan et al. (2005)
M58	Enhancement of seamlessness of interactions	Not defined	Ranganathan et al. (2005)
M59	Ease of information retrieval, versioning and archiving processes measured by user feedback	Not defined	Ranganathan et al. (2005)
M60	Reduction in number of steps that user has to take to get some information or the number of parameters that user has to enter in her/his query	Defined but without measurement function	Ranganathan et al. (2005)
M61	Precision: granularity with which context information describes a real-world situation	Not defined	Damián-Reyes et al. (2011)
M62	Probability of correctness: the probability that an instance of context accurately represents the corresponding real-world situation, as assessed by the context source, at the time it was determined	Defined but without measurement function	Damián-Reyes et al. (2011), Sanchez-pi and Carb (2012)
M63	Sensor precision and accuracy	Not defined	Damián-Reyes et al. (2011)
M64	Performance of the sensor	Not defined	Damián-Reyes et al. (2011)
M65	Accuracy: (True positives + True negatives)/ False positives + False negatives + True positives + True negatives); True positive (tp)—the decision was to disclose, and the prediction was correct; False positives (fp)—the decision was to disclose, and the prediction was incorrect; True negatives (tn)—the decision was to deny, and the prediction was correct; False negatives (fn)—the decision was to deny, and the prediction was incorrect	Well defined	Toch (2011)
M66	Automation: (A/A + B) A: the number of decisions taken autonomically by the decision engine B: the number of decisions sent to the user.	Well defined	Toch (2011)
M67	Adaptation correctness: $\frac{\sum_{i=1}^{N}(Ai/Bi)*100}{N}$ N = number of adaptations Ai = Number of correctly performed adaptations *i* Bi = Number of performed adaptations *i*	Well defined	Santos et al. (2013)

Table 10 continued

ID	Software measure	Classification	References
M68	Context correctness: $\frac{\sum_{j=1}^{N}(Aj/Bj)*100}{N}$ N = Number of different context information Ai = Number of correct context information *j* Bi = Number of collected adaptations *i*	Well defined	Santos et al. (2013)
M69	Context frequency = Frequency of context changing Low = minutes, Medium = seconds, High = millisecond	Well defined	Santos et al. (2013)
Part III			
M70	Adaptation time = the time it takes to adapt Short = millisecond, Medium = seconds, High = minutes	Well defined	Santos et al. (2013)
M70	Accuracy: the probability that a piece of context information is correct. $\mathrm{RMSE(Si)} = \sqrt{\frac{1}{N} * \sum_{i=1}^{N}(xi - x)^2}$ RMSE = root-mean-squared error N is the total of observed data values, X*i* is the observed data value, X is the average of the observed data values	Well defined	Kim and Lee (2006)
M71	Completeness: the extent to which data is not missing and is of sufficient task at hand CS*i* = AD/TD CS*i* is the completeness of a sensor S*i* AD is the number of available output values and the total number of output values.	Well defined	Kim and Lee (2006)
M72	Representation consistency: the extent to which data is presented in the same format	Not defined	Kim and Lee (2006)
M73	Access security: the extent to which access to data is restricted appropriately to maintain its security	Not defined	Kim and Lee (2006)
M74	Up-to-dateness: the extent to which the data is sufficiently up-to-date for the task at hand	Not defined	Kim and Lee (2006)
M75	User ability to change input devices as improvements	Not defined	Ryu et al. (2006)
M76	User ability to sustain the weights of the devices	Not defined	Ryu et al. (2006)
M13	Number of tasks user can accomplish that were not originally envisioned	Defined but without measurement function	Ryu et al. (2006)
M77	Number of hours for the portable artifacts to survive without extra power supply	Defined but without measurement function	Ryu et al. (2006)
M78	Location precision	Not defined	Sanchez-pi and Carb (2012)
M79	Up-to-dateness specifies the age of context information	Not defined	Sanchez-pi and Carb (2012)

Table 10 continued

ID	Software measure	Classification	References
M80	Refresh rate is related to up-to-dateness, and describes how often it is possible or desired to receive a new measurement	Not defined	Sanchez-pi and Carb (2012)

Table 11 Software measures for device capability

ID	Software measure	Classification	References
M81	Device OS	Not defined	De Moor et al. (2010)
M82	CPU utilization	Not defined	De Moor et al. (2010)
M83	Memory consumption	Not defined	De Moor et al. (2010), Zhang et al. (2006)
M84	Battery status	Not defined	Zhang et al. (2006), De Moor et al. (2010)
M85	Screen size	Not defined	Zhang et al. (2006), De Moor et al. (2010)
M86	Color depth	Not defined	Zhang et al. (2006)

Table 12 Software measures for efficiency

ID	Software measure	Classification	References
M87	Time to complete the whole shopping process	Defined but without measurement function	Cappiello et al. (2009)
M88	Number of errors in tag readings	Defined but without measurement function	Cappiello et al. (2009)
M89	Time to complete a task	Defined but without measurement function	Scholtz and Consolvo (2004)
M90	Performance speed: measures of time from user interaction to feedback for user	Defined but without measurement function	Scholtz and Consolvo (2004)

Table 13 Software measures for mobility

ID	Software measure	Classification	References
M91	Degree of coverage by wireless or mobile network	Not defined	Kourouthanassis et al. (2008)
M92	Degree of Quality of Service (QoS)	Not defined	Kourouthanassis et al. (2008)
M93	Capability of application and/or service migration	Not defined	Kourouthanassis et al. (2008)
M94	Device handover rate = A/B, where A = Number of successful device handovers, B = Number of attempts to handover	Well defined	Ryu et al. (2006)
M95	A/T, where A = number of cases encountered by the users with the disconnection in the system beyond allowable	Well defined	Ryu et al. (2006)

Table 14 Software measure for network capabilities

ID	Software measure	Classification	References
M96	Latency	Not defined	Zhang et al. (2006)
M97	Type of current access (GPRS, UMTS, H SPA, LTE, Wi-Fi, WiMAX, DV B-H)	Not defined	De Moor et al. (2010)
M98	The strength of the perceived signal	Not defined	De Moor et al. (2010)
M99	Throughput	Not defined	De Moor et al. (2010)
M100	Packet loss	Not defined	De Moor et al. (2010)
M101	Delay: time taken for a message to be transmitted	Defined but without measurement function	De Moor et al. (2010), Chalmers and Sloman (1999)
M102	Response time: round-trip time from request transmission to reply receipt	Defined but without measurement function	Chalmers and Sloman (1999)
M103	Jitter: variation in delay or response time	Not defined	De Moor et al. (2010), Chalmers and Sloman (1999)
M104	Bandwidth required or available, in bits or bytes per second	Not defined	Zhang et al. (2006), Chalmers and Sloman (1999)
M105	Bandwidth required or available, in application specific units per second, e.g., video frame rate	Not defined	Zhang et al. (2006), Chalmers and Sloman (1999)
M106	Transaction rate: number of operations requested or processed per second	Defined but without measurement function	Chalmers and Sloman (1999)
M107	Picture detail: pixel resolution	Not defined	Chalmers and Sloman (1999)
M108	Picture color accuracy: maps to color information per pixel	Not defined	Chalmers and Sloman (1999)
M109	Video rate: maps to frame rate	Not defined	Chalmers and Sloman (1999)
M110	Video smoothness: maps to frame rate jitter	Not defined	Chalmers and Sloman (1999)
M111	Audio quality: audio sampling rate and number of bits	Not defined	Chalmers and Sloman (1999)
M112	Video/audio synchronization: video and audio stream synchronization, e.g., for lip-sync	Not defined	Chalmers and Sloman (1999)
M113	Per-use cost: cost to establish a connection, or gain access to a resource	Not defined	Chalmers and Sloman (1999)
M114	Per-unit cost: cost per unit time or per unit of data, e.g., connection time charges and per query charges	Not defined	Chalmers and Sloman (1999)

Table 15 Software measures for privacy

ID	Software measure	Classification	References
M115	Type of information user has to divulge to obtain value from application	Not defined	Scholtz and Consolvo (2004)

Table 15 continued

ID	Software measure	Classification	References
M116	Availability of the user's information to other users of the system or third parties	Not defined	Scholtz and Consolvo (2004)
M117	Amount of information a user has to divulge to obtain value from application	Defined but without measurement function	Jafari et al. (2011)
M118	Availability of explanations to a user about the potential use of recorded data	Not defined	Jafari et al. (2011)
M119	User control over private information: 0–3, where 0 = no control provided 1 = system provides control over the disclosure of one kind of information (content, location, or identity) 2 = system provides control over two kinds of information 3 = system provides control over all three kinds of information.	Defined but without measurement function	Jafari et al. (2011)
M120	Expressiveness of the security (privacy) policy: a value of 0–4, representing the number of features supported (1. Support for mandatory and discretionary rules, 2. Context sensitivity, 3. Uncertainty handling, 4. Conflict resolution)	Defined but without measurement function	Jafari et al. (2011)
M121	Alert ratio: % of operations that goes unnoticed. Ideal value = 0	Defined but without measurement function	Jafari et al. (2011)
M122	Choice ratio: % of operations with no option. The higher the number of options the better. Ideal value = 0	Defined but without measurement function	Jafari et al. (2011)
M123	Consent ratio: % of operation which utilizes user information but do not require user consent. Ideal value = 0	Defined but without measurement function	Jafari et al. (2011)
M124	Scenarios counts: the reciprocal of the # of scenarios (policies or rules) that determine what decision to take. The more the number of scenarios the better the system. Ideal value = 0	Defined but without measurement function	Jafari et al. (2011)
M125	Anonymity counts: the reciprocal of the # of user identities that are anonymous. The higher the # the better the system. Ideal value = 0.	Defined but without measurement function	Jafari et al. (2011)
M126	Log index: % of distinct operations that do not have explicit logging or feedback mechanism. Ideal value = 0.	Defined but without measurement function	Jafari et al. (2011)

Table 16 Software measures for reliability

ID	Software measure	Classification	References
M127	A/B, where A = number of cases in which user succeeded to exchange data with other systems, B = number of cases in which user attempted to exchange data	Well defined	Ryu et al. (2006)

Table 16 continued

ID	Software measure	Classification	References
M128	Reliability R(u) = exp(-uλ(x)) = exp(-u/MTTF). Here u is the projected execution time in the future, x is a variable of integration, and λ(x) is the failure rate	Well defined	Ryu et al. (2006)
M129	MTTF = 1/λ(t), The MTTF is the mean time to failure of the software (i.e., the average active time until a failure occurrence)	Well defined	Ryu et al. (2006), Chalmers and Sloman (1999)
M130	Mean Time to Repair (MTTR) = Down time from failure to restarting normal operation	Well defined	Chalmers and Sloman (1999)
M131	Mean Time Between Failures (MTBF) = MTTF + MTTR	Well defined	Chalmers and Sloman (1999)
M132	Percentage of time available = MTTF/ MTTF + MTTR	Well defined	Chalmers and Sloman (1999)
M133	Loss or Corruption rate = Proportion of total data that does not arrive as sent, e.g., network error rate	Well defined	Chalmers and Sloman (1999)

Table 17 Software measures for reversibility

ID	Software measure	Classification	References
M134	Tc-Ts, where Tc = time of completing correction of specified type of errors from performed tasks, and Ts = time of starting correction of specified type of errors from performed tasks	Well defined	Ryu et al. (2006)
M135	A/UOT where A = number of times that the user succeeds to cancel their error operation, UOT = user operating time during observation period	Well defined	Ryu et al. (2006)
M136	A/B, where A = Number of instances where the input data were successfully modified or changed before being elaborated, B = number of instances where user tried to modify or to change the input data during observed user operating time	Well defined	Ryu et al. (2006)
M137	A/B, where A = number of input errors which the user successfully corrects, and B = number of attempts to correct input errors	Well defined	Ryu et al. (2006)

Table 18 Software measures for robustness

ID	Software measure	Classification	References
M138	Percentage of transient faults that were invisible to user	Defined but without measurement function	Scholtz and Consolvo (2004)
M139	Measures of interruptions based on dynamic set of users, hardware, or software	Not defined	Scholtz and Consolvo (2004)

Table 19 Software measure for scalability

ID	Software measure	Classification	References
M140	Effectiveness of interactions with large numbers of entities or users	Not defined	Scholtz and Consolvo (2004)

Table 20 Software measures for security

ID	Software measure	Classification	References
M119	User control over private information: content privacy, identity privacy and location privacy	Not defined	Ranganathan et al. (2005)
M120	Expressiveness of the security policy: support for mandatory and discretionary rules, context sensitivity, uncertainty handling, conflict resolution	Not defined	Ranganathan et al. (2005)
M141	Unobtrusiveness of security mechanisms: % of time used for interacting with the security subsystem (e.g., authentication) auxiliary to the main task	Defined but without measurement function	Ranganathan et al. (2005)

Table 21 Software measures for simplicity

ID	Software measure	Classification	References
M142	Mean time taken to use a function correctly	Defined but without measurement function	Ryu et al. (2006)
M143	Mean user operation time until user achieved to perform the specified task within a short time	Defined but without measurement function	Ryu et al. (2006)
M144	Mean number of steps to activities	Defined but without measurement function	Ryu et al. (2006)
M145	A/B, where A = number of reduced operation procedures after customizing operation, B = number of operation procedures before customizing operation	Well defined	Ryu et al. (2006)

Table 22 Software measures for transparency

ID	Software measure	Classification	References
M146	Usability of interaction modalities and perceived distraction for users	Not defined	Kourouthanassis et al. (2008)
M147	Degree of artifact embedment to the physical space	Not defined	Kourouthanassis et al. (2008)
M148	Degree of conformance or changes evoked to the existing physical architecture	Not defined	Kourouthanassis et al. (2008)
M149	Number and types of interaction modalities and degree of support for interactions through natural interfaces (for example: tangible and speech-based)	Defined but without measurement function	Kourouthanassis et al. (2008)
M150	Effectiveness comparisons on different sets of I/O devices	Not defined	Scholtz and Consolvo (2004), Ryu et al. (2006)

Table 22 continued

ID	Software measure	Classification	References
M151	Perceived transparency of the system	Not defined	Evers et al. (2010)
M152	User's understanding of the system explanation	Not defined	Scholtz and Consolvo (2004)
M153	Effectiveness of interactions provided for user control of system initiative	Not defined	Scholtz and Consolvo (2004)
M154	Match between the system's contextual model and the actual situation	Not defined	Scholtz and Consolvo (2004)
M155	Appropriateness of action	Not defined	Scholtz and Consolvo (2004)
M156	Match between the system action and the action the user would have requested	Not defined	Scholtz and Consolvo (2004)
M157	Time to explicitly enter personalization information	Defined but without measurement function	Scholtz and Consolvo (2004)
M158	Time for the system to learn and adapt to the user's preferences	Defined but without measurement function	Scholtz and Consolvo (2004)
M159	Degree of ambiguity of the application (including commands and dialogues)	Not defined	Ryu et al. (2006)
M160	Using multimodal interactions?	Not defined	Ryu et al. (2006)

Table 23 Software measures for trust

ID	Software measure	Classification	References
M161	Ease of coordination with others in multiuser application	Not defined	Scholtz and Consolvo (2004), Ryu et al. (2006)
M162	Number of collisions with activities of others	Defined but without measurement function	Scholtz and Consolvo (2004), Ryu et al. (2006)
M163	User understanding about how recorded data is used	Not defined	Scholtz and Consolvo (2004)
M164	User understanding inferences that can be drawn about him or her by the application	Not defined	Scholtz and Consolvo (2004)
M165	Ability for users to manage how and by whom their data is used	Not defined	Scholtz and Consolvo (2004)
M166	Types of recourse available to user in the event that his or her data is misused	Not defined	Scholtz and Consolvo (2004)

Table 23 continued

ID	Software measure	Classification	References
M19	Number of events not noticed by a user in acceptable times	Defined but without measurement function	Ryu et al. (2006)
M18	Number of display/actions users need to accomplish an interaction or to check on the progress of an interaction	Defined but without measurement function	Ryu et al. (2006)
M21	Workload imposed on the user attributable to focus	Not defined	Ryu et al. (2006)
M17	Number of times a user must change focus due to technology	Defined but without measurement function	Ryu et al. (2006)

Table 24 Software measures for usability

ID	Software measure	Classification	References
M167	Head turns: total number per task	Defined but without measurement function	Ranganathan et al. (2005)
M168	Physical Movement: % of time used for movement auxiliary to the main task	Defined but without measurement function	Ranganathan et al. (2005)
M169	A priori user knowledge: total number of facts required to be known by the user to perform task	Defined but without measurement function	Ranganathan et al. (2005)
M170	Keystrokes, clicks, and other atomic input: Total number per task	Defined but without measurement function	Ranganathan et al. (2005)
M171	Error and Error Recovery: total number of errors, and time spent recovering from error	Defined but without measurement function	Ranganathan et al. (2005)
M172	Minimum number of user operations necessary to perform a particular function	Defined but without measurement function	Ross and Burnett (2001)
M173	Average time to enter a destination	Defined but without measurement function	Ross and Burnett (2001)
M174	Word Error Rate: (Number of Substitution + Insertions + Deletions)/Total number of words	Well defined	Schalkwyk et al. (2010)
M175	Semantic Quality (WebScore): Number of correct search results/Total number of spoken query	Well defined	Schalkwyk et al. (2010)
M176	Out-of-Vocabulary Rate: percentage of words spoken by the user that are not modeled by the language model. It is important to keep this number as low as possible	Defined but without measurement function	Schalkwyk et al. (2010)
M177	Latency is defined as the total time (in seconds) it takes to complete a search request by voice. More precisely, we define latency as the time from when the user finishes speaking until the search results appear on the screen	Defined but without measurement function	Schalkwyk et al. (2010)

Table 24 continued

ID	Software measure	Classification	References
M178	Perplexity is crudely speaking, a measure of the size of the set of words that can be recognized next, given the previously recognized words in the query	Defined but without measurement function	Schalkwyk et al. (2010)

Table 25 Software measure for user satisfaction

ID	Software measure	Classification	Reference
M179	User rating of performing the task	Not defined	Scholtz and Consolvo (2004)
M180	Enjoyment level when using the application	Not defined	Scholtz and Consolvo (2004)
M181	Level of anticipation prior to using the application	Not defined	Scholtz and Consolvo (2004)
M182	Sense of loss when the application is unavailable	Not defined	Scholtz and Consolvo (2004)
M183	Aesthetics: ratings of application look and feel	Not defined	Scholtz and Consolvo (2004)
M184	Status: pride in using and owning the application	Not defined	Scholtz and Consolvo (2004)
M185	Peer pressure felt to use or own the application	Not defined	Scholtz and Consolvo (2004)
M186	User satisfaction subjective (1–5) scaling (5 = most agreement)	Defined but without measurement function	Ranganathan et al. (2005)

Table 26 Software measures for utility

ID	Software measure	Classification	References
M187	Changes in productivity or performance	Not defined	Scholtz and Consolvo (2004)
M188	Changes in output quality	Not defined	Scholtz and Consolvo (2004)
M189	Behavior changes: type, frequency, and duration	Not defined	Scholtz and Consolvo (2004)
M190	Willingness to modify behavior or tasks to use application;	Not defined	Scholtz and Consolvo (2004)
M191	Comfort ratings of wearable system components	Not defined	Scholtz and Consolvo (2004)
M192	Requirements placed on user outside of social norms	Not defined	Scholtz and Consolvo (2004)
M193	Aesthetic ratings of system components	Not defined	Scholtz and Consolvo (2004)
M194	Perceived usefulness	Not defined	Evers et al. (2010)

Table 27 Software measures without characteristic

ID	Software measure	Classification	References
Part I			
M195	Agreement degree of service functions: Service concordance (SC): A (number of expected and comprehended service functions)/B (number of functions provided from ubiquitous service	Well defined	Lee and Yun (2012), Lee et al. (2008)
M196	User's participating degree for bidirectional communication of ubiquitous service while using service	Not defined	Lee and Yun (2012), Lee et al. (2008)
M197	User immersion degree in the ubiquitous service without own location-awareness, while using the service	Not defined	Lee and Yun (2012), Lee et al. (2008)
M198	The degree of understanding input data and expecting output at service request: A (number of expectable and performable I/O)/B (number of I/O provided from ubiquitous service)	Well defined	Lee and Yun (2012), Lee et al. (2008)
M199	User's approved time before using the service	Defined but without measurement function	Lee et al. (2008)
M200	Learning time to use new service functions	Defined but without measurement function	Lee et al. (2008)
M201	Time of user spending in hesitation or on hold to use the service	Defined but without measurement function	Lee et al. (2008)
M202	Number of user out of controls during service use	Defined but without measurement function	Lee et al. (2008)
M203	Predictability of application behavior: Degree of match between user model and behavior of application	Not defined	Scholtz and Consolvo (2004)
M204	Degree of match between user's model and actual functionality of the application	Not defined	Scholtz and Consolvo (2004)
M205	Degree of match between user's understanding of his or her responsibilities, system responsibilities, and the actual situation	Not defined	Scholtz and Consolvo (2004)
M206	Degree to which user understands the application's boundary	Not defined	Scholtz and Consolvo (2004)
M207	Vocabulary awareness: degree of match between user's model and the syntax used by the application	Not defined	Scholtz and Consolvo (2004)
Part II			
M208	Usable input method or various interactivity degrees (keyboard, stylus, audio input, touch screen) = Mean number or Degree = Type, applicable degree of multimodal interface	Defined but without measurement function	Lee and Yun (2012)
M209	Usable degree of multimodal devices same service functions = Applicable degree of same function with multimodal interface	Not defined	Lee and Yun (2012)

Table 27 continued

ID	Software measure	Classification	References
M210	Usable range of requesting the service (sensing distance): Dc = Successful distance based on user's location—recognizable degree of use for the usable distance	Not defined	Lee and Yun (2012)
M211	Suitability ratio of service functions: number of problem occurrence or number of total functions	Defined but without measurement function	Lee and Yun (2012)
M212	Data adjustable degree between service in operating and the component of other services	Not defined	Lee and Yun (2012)
M213	Acknowledgment degree of service defense system against trespassing of other user	Not defined	Lee and Yun (2012)
M214	Instinctive understandability degree of the service function result	Not defined	Lee and Yun (2012)
M215	Frequency or number of service system error during using service	Defined but without measurement function	Lee and Yun (2012)
M216	Number of requesting help to confirm the specific output device result	Defined but without measurement function	Lee and Yun (2012)
M217	Suitability degree of service feedback felt by user	Not defined	Lee and Yun (2012)
M218	Required time until completion of loading specific service function	Defined but without measurement function	Lee and Yun (2012)

References

Abi-Char, P. E., Mhamed, A., El-Hassan, B., & Mokhtari, M. (2010). A flexible privacy and trust based context-aware secure framework. In *Proceedings of the aging friendly technology for health and independence, and 8th international conference on smart homes and health telematics*, ICOST'10. Springer. http://dl.acm.org/citation.cfm?id=1894439.1894443.

Ammar, L., Trabelsi, A., & Mahfoudhi, A. (2015). A model-driven approach for usability engineering of interactive systems. *Software Quality Journal*, 1–35.

Bezerra, C. I. M., Oliveira, K. M., Andrade, R. M. C., et al. (2014). Challenges for usability testing in ubiquitous system. In *l'Interaction Homme-Machine*.

Cappiello, I., Puglia, S., & Vitaletti, A. (2009). Design and initial evaluation of a ubiquitous touch-based remote grocery shopping process. In *First international workshop on near field communication*, 9–14.

Carvalho, R. M., Andrade, R. M. C., & Oliveira, K. M. (2015). Using the GQM method to evaluate calmness in ubiquitous applications*. In *HCI international*.

Chalmers, D., & Sloman, M. (1999). A survey of quality of service in mobile computing environments. *IEEE Communications Surveys & Tutorials, 2*(2), 2–10.

Chang, Y.-H., & Lin, B.-S. (2011). An inquiry-based ubiquitous tour system. In *International conference on complex, intelligent and software intensive systems*.

Da Silva, C. M. R., Da Silva, J. L. C., Rodrigues, R. B., Do Nascimento, L. M., & Garcia, V. C. (2013). Systematic mapping study on security threats in cloud computing. arXiv preprint arXiv:1303.6782,

Damián-Reyes, P., Favela, J., & Contreras-Castillo, J. (2011). Uncertainty management in context-aware applications: Increasing usability and user trust. *Wireless Personal Communications, 56*(1), 37–53.

De Moor, K., Ketyko, I., Joseph, W., et al. (2010). Proposed framework for evaluating quality of experience in a mobile, testbed-oriented living lab setting. *Mobile Networks and Applications, 15*(3), 378–391.

Dey, A. K. (2001). Understanding and using context. *Personal and Ubiquitous Computing, 5*(1), 4–7.

Evers, V., Cramer, H., Van Someren, M., & Wielinga, B. (2010). Interacting with adaptive systems. In *Interactive collaborative information systems* (pp. 299–325). Springer.

Evers, C., Kniewel, R., Geihs, K., & Schmidt, L. (2014). The user in the loop: Enabling user participation for self-adaptive applications. *Future Generation Computer Systems, 34*, 110–123.

Fenton, N., & Pfleeger, S. (1997). Software metrics: A rigorous and practical approach. *PWS Pub*.

Haapalainen, E., Kim, S., Forlizzi, J. F., & Dey, A. K. (2010). Psycho-physiological measures for assessing cognitive load. In *Proceedings of the 12th ACM international conference on ubiquitous computing*, Ubicomp'10. ACM. http://doi.acm.org/10.1145/1864349.1864395.

Hernandes, E., Zamboni, A., Fabbri, S., & Thommazo, A. Di. (2012). Using GQM and TAM to evaluate StArt—A tool that supports systematic review. *CLEI Electronic Journal, 15*, 3.

Iqbal, R., Sturm, J., Kulyk, O., Wang, J., & Terken, J. (2005). User-centred design and evaluation of ubiquitous services. In *Proceedings of the 23rd international conference on design of communication—documenting and designing for pervasive information*. http://www.scopus.com/inward/record.url?eid=2-s2.0-32044471829&partnerID=40&md5=9a711a41b59e2b258fd13669a3826bc7.

ISO 9241-11. (1998). Ergonomic requirements for office work with visual display terminals. In *The International Organization for Standardization*

ISO/IEC 14598. (1999). Information Technology—Software Product Evaluation—Part 1.

ISO/IEC 9126. (2001). Software engineering—Product Quality—Part 1.

ISO/IEC 25000. (2014). Software Engineering—Software Product Quality Requirements and Evaluation (SQuaRE)—Guide to SQuaRE.

ISO/IEC 25010. (2011). ISO/IEC 25010. *Systems and software engineering—Systems and software Quality Requirements and Evaluation (SQuaRE)—System and software quality models*, v. 2011.

Jafari, S., Mtenzi, F., O'Driscoll, C., Fitzpatrick, R., & O'Shea, B. (2010). Privacy metrics in ubiquitous computing applications. In *International conference for internet technology and secured transactions*.

Jafari, S., Mtenzi, F., O'Driscoll, C., Fitzpatrick, R., & O'Shea, B. (2011). Measuring privacy in ubiquitous computing applications. *International Journal of Digital Society, 2*(3), 547–550.

Jia, L., Collins, M., & Nixon, P. (2009). Evaluating trust-based access control for social interaction. In *3rd international conference on mobile ubiquitous computing, systems, services, and technologies*. http://www.scopus.com/inward/record.url?eid=2-s2.0-77951440915&partnerID=40&md5=241ed20a5c57e71b94e32fcaec543a9e.

Karaiskos, D., Kourouthanassis, P., & Giaglis, G. M. (2009). Towards a validated construct for information systems pervasiveness: An exploratory assessment. In *BLED 2009 proceedings*, p. Paper 12.

Karvonen, H., & Kujala, T. (2014). Designing and evaluating ubicomp characteristics of intelligent in-car systems. In *5th International conference on applied human factors and ergonomics*.

Kemp, E. A., Thompson, A.-J., & Johnson, R. S. (2008). Interface evaluation for invisibility and ubiquity: An example from E-learning. In *Proceedings of the 9th ACM SIGCHI New Zealand Chapter's international conference on human–computer interaction: Design Centered HCI*, CHINZ'08. ACM. http://doi.acm.org/10.1145/1496976.1496981.

Kim, H. J., Choi, J. K., & Ji, Y. (2008). Usability evaluation framework for ubiquitous computing device. In *Proceedings—3rd international conference on convergence and hybrid information technology, ICCIT 2008*. http://www.scopus.com/inward/record.url?eid=2-s2.0-57849159822&partnerID=40&md5=2965dfe3adcf63a35f173ca1fd4cb7e9.

Kim, Y., & Lee, K. (2006). A quality measurement method of context information in ubiquitous environments. In *2006 International conference on hybrid information technology*. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4021269.

Kitchenham, B. A., Budgen, D., & Brereton, P. (2010). The value of mapping studies—A participant-observer case study. In *Proceedings of evaluation and assessment of software engineering, EASE*.

Kitchenham, B. A., & Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. In *Technical report. EBSE-2007-01, Keele University*.

Ko, I.-Y., Koo, H.-M., & Jimenez-Molina, A. (2010). User-centric web services for ubiquitous computing. In *Advanced techniques in web intelligence-I*.

Kourouthanassis, P. E., Giaglis, G. M., & Karaiskos, D. C. (2008). Delineating the degree of "pervasiveness" in pervasive information systems: An assessment framework and design implications. In *Pan-Hellenic conference on informatics, PCI*.

Kryvinska, N., Strausss, C., & Zinterhof, P. (2011). "Variated availability" approach to the services manageable delivering. In *Fifth international conference on innovative mobile and internet services in ubiquitous computing (IMIS)*.

Lee, J., & Yun, M. H. (2012). Usability assessment for ubiquitous services: Quantification of the interactivity in inter-personal services. In *IEEE international conference on management of innovation & technology*.

Lee, J., Song, J., Kim, H., Choi, J., & Yun, M. H. (2008). A user-centered approach for ubiquitous service evaluation: An evaluation metrics focused on human–system interaction capability. In *Asia-Pacific conference, APCHI*.

Liampotis, N., Roussaki, I., Papadopoulou, E., et al. (2009). A privacy framework for personal self-improving smart spaces. In *International conference on computational science and engineering*.

Montagud, S., Abrahão, S., & Insfran, E. (2012). A systematic review of quality attributes and measures for software product lines. In *Software Quality Journa*, v. 20.

Nielsen, J. (1994). *Usability engineering*. Access Online via Elsevier.

Novais, R. L., Torres, A., Mendes, T. S., Mendonça, M., & Zazworka, N. (2013). Software evolution visualization: A systematic mapping study. *Information and Software Technology, 55*(11), 1860–1883.

Oriol, M., Marco, J., & Franch, X. (2014). Quality models for web services: A systematic mapping. *Information and Software Technology, 56*(10), 1167–1182.

Petersen, K., Feldt, R., Mujtaba, S., & Mattsson, M. (2008). Systematic mapping studies in software engineering. In *Proceedings of the 12th international conference on evaluation and assessment in software engineering*, EASE'08. British Computer Society. http://dl.acm.org/citation.cfm?id=2227115.2227123.

Petersen, K., & Gencel, C. (2013). Worldviews, research methods, and their relationship to validity in empirical software engineering research. In *Proceedings—Joint conference of the 23rd international workshop on software measurement and the 8th international conference on software process and product measurement, IWSM-MENSURA 2013* (pp. 81–89).

Petersen, K., Vakkalanka, S., & Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology, 64*, 1–18.

Poppe, R., Rienks, R., & Dijk, B. Van (2007). Evaluating the future of HCI: Challenges for the evaluation of emerging applications. *AI for Human Computing*, 234–250.

Ranganathan, A., Al-Muhtadi, J., Biehl, J., et al. (2005). Towards a pervasive computing benchmark. In *International conference on pervasive computing and communications workshops*.

Reis, R. A. C. (2015). Usability evaluation approaches for (ubiquitous) mobile applications: A systematic mapping study usability evaluation approaches for (ubiquitous) mobile applications: A systematic mapping study. n. September.

Rocha, L. S., Ferreira, J. B., Lima, F. F. P., et al. (2011). Ubiquitous software engineering: Achievements, challenges and beyond. In *Brazilian symposium on software engineering* (**in Portuguese**)

Ross, T., & Burnett, G. (2001). Evaluating the human–machine interface to vehicle navigation systems as an example of ubiquitous computing. In *International Journal of Human–Computer Studies*,

Rubio, J. M. L., & Bozo, J. P. (2007). Approach to a quality process for the ubiquitous software development. In *Electronics, robotics and automotive mechanics conference.*

Ryu, H., Hong, G. Y., & James, H. (2006). Quality assessment technique for ubiquitous software and middleware. *Research Letters in the Information and Mathematical Sciences, 9*, 13–87.

Sanchez-pi, N., & Carb, J. (2012). An evaluation method for context—Aware systems in U-Health. In *3rd international symposium on ambient intelligence (ISAmI 2012)*.

Santos, R. M., Oliveira, K. M., Andrade, R. M. C., Santos, I. S., & Lima, E. R. R. (2013). A quality model for human–computer interaction evaluation in ubiquitous systems. In *Latin American conference on human computer interaction*.

Schalkwyk, J., Beeferman, D., Beaufays, F., et al. (2010). Your word is my command: Google Search by Voice: A case study. *Advances in Speech Recognition*.

Scholtz, J., & Consolvo, S. (2004). Toward a framework for evaluating ubiquitous computing applications. *IEEE Pervasive Computing*,

Sears, A., & Jacko, J. A. (2009). *Human–computer interaction: Development process*. Boca Raton: CRC Press.

Silveira, P. A. M., Machado, I. C., McGregor, J. D., Santana, E., & Meira, S. R. L. (2011). A systematic mapping study of software product lines testing. In *Information and Software Technology*

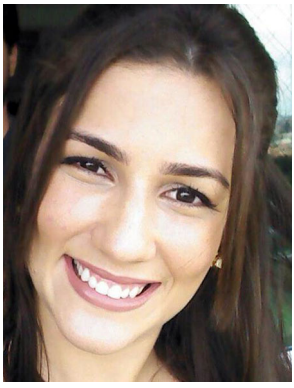
Song, J., Park, K. R., Kwon, S., Lee, J. H. J. H., & Yun, M. H. (2009). The development of human-system interactivity metrics for ubiquitous service applying user-centered design methodology. In *World Congress on Services*.

Sousa, B., Pentikousis, K., & Curado, M. (2011). UEF: Ubiquity evaluation framework. *Wired/Wireless Internet Communications*

Spínola, R. O., & Travassos, G. H. (2012). Towards a framework to characterize ubiquitous software projects. *Information and Software Technology*

Sun, T., & Denko, M. K. (2008). Performance evaluation of trust management in pervasive computing. In *International conference on advanced information networking and applications, AINA*.

Tahir, T., & Jafar, A. (2011). A systematic review on software measurement programs. In *Frontiers of Information Technology (FIT), 2011* (Vol. 73, pp. 39–44).
Thompson, S. G., & Azvine, B. (2004). No pervasive computing without intelligent systems. In *BT technology journal*
Toch, E. (2011). Super-Ego: A framework for privacy-sensitive bounded context-awareness. In *ACM international workshop on context-awareness for self-managing systems*.
Viana, J. R. M., Viana, N. P., Trinta, F. A. M., & Carvalho, W. V. De (2014). A systematic review on software engineering in pervasive games development. In *Brazilian symposium on computer games and digital entertainment*. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7000032.
Wagner, S., Toftegaard, T., & Bertelsen, O. (2012). Requirements for an evaluation infrastructure for reliable pervasive healthcare research. In *International conference on pervasive computing technologies for healthcare*, IEEE.
Waibel, A., Stiefelhagen, R., Carlson, R., et al. (2010). Computers in the human interaction loop. *Handbook of Ambient Intelligence and Smart Environments*.
Weihong-Guo, A., Blythe, P., Olivier, P., Singh, P., & Nam Ha, H. (2008). Using immersive video to evaluate future traveller information systems. In *IET intelligent transport systems*,
Weiser, M. (1991). The computer for the 21st century. In *Scientific American*.
Wieringa, R., Maiden, N., Mead, N., & Rolland, C. (2005). Requirements engineering paper classification and evaluation criteria: A proposal and a discussion. In *Requirements Engineering.*
Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *18th international conference on evaluation and assessment in software engineering (EASE 2014)* (pp. 1–10).
Wohlin, C., Runeson, P., Da Mota Silveira Neto, P. A., et al. (2013). On the reliability of mapping studies in software engineering. *Journal of Systems and Software, 86*(10), 2594–2610.
Wu, C. L., & Fu, L. C. (2012). Design and realization of a framework for human–system interaction in smart homes. *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, *42*(1), 15–31.
Zhang, Y., Zhang, S., Tong, H., & Yong Zhang, S. Z. (2006). Adaptive service delivery for mobile users in ubiquitous computing environments. In *International conference on Ubiquitous Intelligence and Computing*.

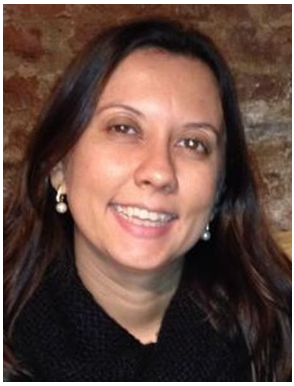


Rainara Maia Carvalho is a PhD student at Federal University of Ceará, Brazil. She received her master's degree in Computer Science from Federal University of Ceará. Her main research areas are ubiquitous computing, software engineering and human–computer interaction focusing on software quality measures.

Rossana Maria de Castro Andrade is an associate professor at Federal University of Ceará, Brazil, in the Department of Computer Science. She received her PhD from School of Information Technology and Engineering of the University of Ottawa, Canada. Her research interests are computer networks and software engineering, specifically ubiquitous systems and software reuse.

Káthia Marçal de Oliveira is an associate professor at University of Valenciennes, France. She has a PhD in software engineering focused on quality assurance. She has worked on the definition of quality measures for different domain applications (e.g., legacy system, web system, software process). Her interests include software quality assurance, knowledge management and measurement.

Ismayle de Sousa Santos is a PhD student in the Federal University of Ceará, Brazil. He received his master's degree in Computer Science from the Federal University of Ceará, in 2013. His main research areas are software quality, software testing and software product line.



Carla Ilane Moreira Bezerra is an assistant professor in the Federal University of Ceará (UFC), Brazil. She received her master's degree in Applied Informatics, in 2009. Currently, she is a PhD student in Computer Science at UFC. Her research interests are software product line, software quality and context-aware computing.