A systematic mapping review of usability evaluation methods for software development process

Article · January 2016

DOI: 10.14257/ijseia.2016.10.1.16

CITATIONS
50

READS
1,320

2 authors, including:

Freddy Paz
Pontifical Catholic University of Peru
52 PUBLICATIONS **217** CITATIONS

SEE PROFILE

International Journal of Software Engineering and Its Applications
Vol. 10, No. 1 (2016), pp. 165-178
http://dx.doi.org/10.14257/ijseia.2016.10.1.16

A Systematic Mapping Review of Usability Evaluation Methods for Software Development Process

Freddy Paz[1] and José Antonio Pow-Sang[1]

[1]*Pontificia Universidad Católica del Perú,*
Av. Universitaria 1801, San Miguel, Lima 32, Perú
fpaz@pucp.pe, japowsang@pucp.edu.pe

Abstract

Given that usability is one of the most important aspects of software quality, several methods have been developed in order to establish techniques capable of evaluating this attribute from early phases of the software development process. However, the choice of the most appropriate method for a particular scenario is still a difficult decision, due to the existence of a vast number of approaches that are described in the literature for this purpose. Therefore, a systematic mapping review was conducted in order to identify the most commonly used usability evaluation techniques in software developments. A total of 1169 studies were identified, of which only 215 studies were selected for this review. According to the analysis, most cases studies establish the use of usability questionnaires as assessment tool. In addition, health informatics and Web applications are the software domain and type of application that are frequently reported in these evaluations. This work has allowed to reach promising results in this area. It is intended to be a guide for specialists to support the choice of the most suitable method for a particular scenario.

Keywords: *Human-Computer Interaction, systematic mapping review, usability evaluation methods, software development process, user-centered design*

1. Introduction

Nowadays, usability is considered one of the most important aspects for the success of any technological product. In the context of software, if a product is difficult to use or provides mechanisms that are hard to understand, then the application is expected to fail [15]. Given the current domain, in which there are several alternatives available for any software product, users will only have a preference for a software capable of supporting them to achieve their goals with satisfaction. For this reason, usability has become highly relevant, especially, during all phases of the software development process [8]. Developers are aware that only a small percentage of users spend their time reading a manual. Therefore, the design of a graphical interface should be intuitive enough to meet users' expectations regarding usability.

The importance of this quality attribute has led to the development of several usability evaluation methods, whose purpose is to determine systematically the degree to which a software product is easy to use. However, because of a broad range of these techniques, the choice of the most suitable method for a particular scenario has become a difficult decision. There is no agreement among specialists on what the best method is. The arrival of numerous techniques has resulted in an extensive discussion to determine the most widely accepted method by the scientific community. In addition, new variants based on the traditional proposals are currently being reported in the literature, without a concrete evidence that these approaches are still employed as procedures to evaluate the usability of software products [11]. This fact justifies the execution of a study to identify the most relevant usability evaluation methods at present.

ISSN: 1738-9984 IJSEIA

The aim of this study was to examine all the case studies that were lately described in the literature to determine the main usability evaluation methods. In a previous work [10], we conducted a preliminary review of the existing techniques that were available for this purpose. In this paper, we present an extended, improved and updated systematic review to determine the current trends in the use of usability evaluation methods for software development contexts. An analysis of the results has allowed to identify the most commonly used techniques for each category of software. These findings are intended to serve as a guide for specialists to support the right choice of an appropriate method in a particular scenario.

This paper is structured as follows. In Section 2, we describe the main concepts that are used in Human-Computer Interaction for studies of this nature. In Section 3, we present the methodology that was used to conduct this study. In Section 4, we discuss the results of our research. Finally, the conclusions and future works are established in Section 5.

2. Background

2.1. Usability

In Software Engineering, the term "usability" is related to the ease of use of a software product. However, this concept can be applied to any technological interface that allows interaction between humans and machines. According to the ISO 9241-11 standard [12], usability can be defined as "the extent to which any product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use". Although this definition is not especially related to software products, the relevance of the user's experience is highlighted.

In the domain of Computer Science, the ISO 9126-1 standard [13] provides a more specialized definition where usability is considered as an attribute of software quality. The concept of usability is defined as "the capability of a software product to be understood, liked, used and attractive when it is used under specified conditions". This definition emphasizes not only the relevance of an aesthetic and intuitive graphical interface but also the potential of a specific software to meet user's expectations.

2.1. Usability Evaluation Methods

Given the relevance of usability in the context of the software development process, several evaluation methods have emerged. The purpose of these techniques is to establish, in a systematic way, the level of usability of graphical software interfaces [4].

According to Fernandez *et. al.*, [2], the usability evaluation methods can be defined as "procedures composed by a series of well-defined activities to collect data related to the interaction between the end user and a software product, in order to determine how the specific properties of a particular software contribute to achieving specific goals". Usually, these methods are employed during all phases of the software development process to ensure the design of a usable product that can meet high-quality standards.

3. Conducting the Systematic Mapping Review

A systematic literature review is a method to analyze, evaluate and interpret all relevant studies to a particular research question, or specific area, or phenomenon of interest. A systematic mapping review is a variant of this technique in which the evidence is plotted at a high level of granularity. Although the SLR is widely used in Medicine, there are proposals to use this methodology in the field of Software Engineering. Kitchenham and Charters [5] establish a set of guidelines to conduct reliable, rigorous and auditable systematic studies for software engineering topics. This protocol defines well-defined procedures to identify and summarize all existing information about a subject

matter in a thorough and unbiased manner. This work was performed according to the parameters defined by Kitchenham and Charters [5]. The steps of this methodology are presented in the subsequent sections.

3.1. Research Questions

The purpose of this work was to determine the current trends in the use of usability evaluation methods for software development processes. Additionally, we identified the software categories that are frequently reported as part of these usability inspections, and the techniques that are commonly employed for each type. In this way, we formulated the following research questions:

RQ1: What are the most widely used techniques to evaluate the usability of software products in the context of a development process?

RQ2: What usability evaluation methods are commonly used for each category of software application involved in a development process?

In order to conduct this review, we defined the general concepts based on PICOC. Since our research is not intended to compare interventions, the "comparison" criterion was not considered. These concepts are detailed in Table I.

Table 1. Definition of the General Concepts Using PICOC

Criterion	Description
Population	Software products.
Intervention	Usability evaluation methods.
Outcomes	Case studies where a usability evaluation is considered as part of a software development process.
Context	Academic context, software industry and all kinds of empirical studies.

3.2. Search Strategy

We defined our search strategy based on the general concepts. Some synonymous were selected to achieve a more comprehensive search. We only considered relevant studies, whose publication date was since 2012, in order to analyze the trends and the current state of art in this field. The resulting string was:

("software development" **OR** "software construction" **OR** "software project" **OR** "software projects" **OR** "software process" **OR** "software processes" **OR** "software engineering" **OR** "software testing" **OR** "software design" **OR** "software verification" **OR** "software validation") **AND** ("method*" **OR** "technique*" **OR** "process*" **OR** "procedure*" **OR** "approach*") **AND** ("test*" **OR** "evaluation*" **OR** "inspection*" **OR** "assessment*" **OR** "measurement*" **OR** "study" **OR** "studies") AND ("usability" **OR** "usable") **AND** (publication year > 2011)

3.3. Search Process

The search process was performed by using three recognized databases to search for primary studies: SCOPUS, ACM Digital Library and ISI (considering Web of Science & Web of Knowledge). No additional study was considered. Once the papers were retrieved, we used Google Scholar to determine the relevance of each paper. Grey literature was excluded since it is not peer reviewed.

3.4. Selection of Primary Studies

Each study that was retrieved from the automated search, was examined by all the authors in order to determine its inclusion in this systematic mapping study. The process of evaluation involved a review of the entire document: title, abstract, introduction, background, state of the art, methodology, study case, results and conclusions. One of

these sections should met the following inclusion criteria: *the study should report at least one usability evaluation applied to a software product in the context of a development process*.

In the same way, we established the exclusion criteria. The studies that met at least one of the following conditions were excluded from this research: (1) *the usability study is not applied to a software product*, and (2) *the usability study is not conducted in a software development context*.

3.5. Data Extraction

We developed a template to register all relevant information about each resulting paper. The data extraction process included the following information: (a) Paper ID, (b) Paper Title, (c) Author(s), (d) Type of publication, (e) Name of the Conference or Journal in which the study was presented, (f) Year of publication, (g) Extraction Date, (h) Database in which the study was found.

The automated search for our systematic mapping review was performed on June 20th, 2015. We obtained 1169 studies from the three consulted databases. After the application of the inclusion and exclusion criteria, 215 of these papers were selected for the review process. Table 2 shows the details regarding the amount of studies that were found during the search process.

Table 2. Summary of Search Results

Database Name	Search Results	Duplicated Papers	Relevant Papers
SCOPUS	488	-	136
ACM	101	75	4
ISI (Journals and Proceedings)	580	247	75
TOTAL	**1169**	**322**	**215**

4. Data Analysis and Results

In order to determine the most used usability evaluation methods, we identified the number of times each technique was reported in relevant papers. All results are summarized in Table 3.

Table 3. Frequency of Use of Each Usability Evaluation Method

Usability Evaluation Method	Number of times the method was used	Percentage (%)
Survey / Questionnaire	104	26.26%
User Testing	56	14.14%
Heuristic Evaluation	50	12.63%
Interview	41	10.35%
User Testing – Thinking Aloud / Thinking Out Loud	38	9.60%
Software Metrics / Usability Metrics	19	4.80%
Automated Evaluation via Software Tool	16	4.04%
Cognitive Walkthrough	11	2.78%
Prototype Evaluation	11	2.78%
Focus Group	6	1.52%
Checklist Verification	5	1.26%
Pencil & Paper	5	1.26%
Others	34	8.59%
TOTAL	**396**	**100.00%**

Table 4. Studies that Report the Use of Each Usability Evaluation Method

Usability Evaluation Method	Studies that Report the Use of the Method (see Appendix A.)
Survey / Questionnaire	001, 002, 005, 006, 009, 015, 017, 019, 020, 023, 034, 037, 040, 043, 046-1, 046-2, 048, 051, 052, 055, 056, 057, 058, 059, 062, 063, 064, 066, 068, 069, 070, 074, 076, 077, 078, 080, 081, 083, 085, 087, 094, 096, 097, 099, 102, 103, 105, 106, 108, 111, 113, 114, 117, 118, 119, 121, 123, 124-2, 126, 127, 131, 134, 136, 137, 140, 141, 142, 144, 145, 147, 149, 151, 154, 158, 160, 161, 167, 169, 170, 174, 175, 176, 177, 178, 179, 181, 182, 185, 188, 190, 192, 194, 198, 199, 201, 202, 206-1, 206-2, 207, 208, 209, 211, 213, 214
User Testing	007, 027, 032, 034, 043, 047, 048, 050, 055, 056, 057, 067, 070, 072, 073, 076, 077, 081, 085, 088, 089, 094, 101, 102, 114, 123, 124-1, 124-2, 125, 127, 129, 132, 133, 138, 139, 140, 146, 150, 153, 158, 160, 161, 165, 176, 179, 181, 183-1, 183-2, 192, 193, 204, 205, 207, 208, 209, 214
Heuristic Evaluation	002, 008, 011-1, 011-2, 012-1, 012-2, 024, 031-1, 031-2, 037, 041, 042, 043, 044, 045, 049, 050, 054, 056, 060, 061, 065, 068, 070, 075, 077, 079, 082, 086, 090, 114, 117, 128, 132, 133, 140, 143, 156, 157, 158, 161, 171, 172, 180, 183-1, 195, 205, 206-1, 206-2, 215
Interview	003, 007, 014, 015, 023, 028, 030, 036, 038, 039, 046-1, 046-2, 051, 052, 055, 058, 063, 071, 081, 083, 095, 105, 110, 111, 119, 123, 127, 138, 140, 144, 146, 156, 161, 163, 173, 174, 182, 183-2, 186, 204, 207
User Testing – Thinking Aloud / Thinking Out Loud	002, 003, 004, 005, 014, 019, 028, 029, 036, 037, 039, 040, 046-1, 046-2, 052, 063, 068, 074, 082, 083, 093-1, 093-2, 095, 099, 104, 109, 117, 118, 134, 156, 159, 162, 173, 174, 182, 184, 188, 197
Software Metrics / Usability Metrics	010, 011-1, 016-1, 016-2, 016-3, 022, 026, 070, 077, 099, 100, 108, 113, 114, 124-2, 155, 178, 200, 202
Automated Evaluation via Software Tool	013, 016-1, 016-2, 016-3, 025, 053, 091, 108, 112, 135, 145, 189, 191, 196-1, 196-2, 196-3
Cognitive Walkthrough	008, 031-1, 031-2, 033, 035, 061, 117, 118, 122, 140, 156
Prototype Evaluation	007, 038, 050, 110, 120, 121, 156, 183-1, 183-2, 190, 193
Focus Group	015, 050, 074, 107, 111, 183-1
Checklist Verification	084, 098, 121, 168, 212
Pencil & Paper	037, 039, 144, 165, 192
Perspective Based Usability Inspection	130-1, 130-2, 206-1, 206-2
Field Observation / Field Study	095, 105, 111
Eye Tracking	052, 164, 166
Click Map / Scroll Map / Heat map	014, 160
Opinion Mining	021, 148
Web Usability Evaluation Process	012-1, 012-2
Retrospective Thinking Aloud	115, 198
Cognitive Task Analysis	051, 165
Usability Guidelines	187
Card Sorting	183-2
Canvas Card Sorting	165
Retrospective Sense Making	162
Personas	165
User Workflow	165
Cognitive Jogthrough	152
Domain Specific Inspection	172
Participatory Heuristic Evaluation	116
Semiotic Inspection Method	210
Usability & Communicability Evaluation Method	092
Simplified Pluralistic Walkthrough	203
Simplified Streamlined Cognitive Walkthrough	203
Music Performance Measurement Method	018

In Table 4, we specify the papers that reported the use of each specific usability evaluation method. However, some studies described using more than one single method. For this reason, there is a difference between the number of primary paper that were identified and the number of usability evaluations that were found. In order to distinguish these studies from references, a three-digit number was assigned to each paper. This list of studies can be found in Appendix A. In the case of papers 011, 012, 016, 031, 046, 093, 124, 183, 196 and 206, more than one software product was considered for a usability assessment. The number next to the ID specifies a different software that was evaluated in the same study case.

4.1. Usability Evaluation Methods

In this section, we detail the usability evaluation methods that were found through the execution of the present systematic mapping review. According to Nielsen [14], Zhang *et. al.*, [21], Paz *et. al.*, [9], and Otaiza *et. al.*, [18] these methods can be defined as:

Survey / Questionnaire: It is a list of questionnaire items that representative users have to answer according to a Likert scale. Each statement of the survey is intended to measure a particular usability aspect of the software system or a specific dimension of the user's satisfaction.

User Testing: A representative amount of end users interacts with the software following a list of pre-defined tasks. Exhaustive observations of these human-system interactions allow the identification of usability issues related to the system. This evaluation method is commonly applied in a usability lab whose equipment enables the recording of user's gestures and user's computer screen for later analysis.

Heuristic Evaluation: A group of usability specialists judge whether each dialogue element of the software system follows established usability principles, called "heuristics".

Interview: Both end user and usability specialist participate in a discussion session about the usability of a software application.

User Testing – Thinking Aloud / Thinking Out Loud: This version of user testing involves the execution of the "thinking aloud protocol". Users have to verbalize their thoughts while they interact with the software system. Supervisors should encourage end-users to express their opinions during the activity. In some cases, this indication is only requested at the beginning of the testing.

Usability Metrics / Software Metrics: The purpose of this method is to establish quantitative measurements. Usability metrics quantify the usability of a system regarding effectiveness, efficiency and satisfaction. Usually, some equations are used to determine numeric values about the usability of a system. The participation of a representative number of users is required to generalize the obtained results.

Automated Evaluation via Software Tool/Software: A software tool is used to perform all the activities that are required in a usability evaluation. Depending on the type of software, this tool can be able to simulate human actions. Other applications only keep track of the user's activities, and perform metric-based measurements. Additionally, these systems can generate a log file that can be analyzed after the testing.

Cognitive Walkthrough: A usability specialist simulates the actions of a novice user of the system. During this interaction, the inspector has to identify potential issues of usability.

Prototype Evaluation: Both, an end users and usability specialists participate of a meeting in which users are asked to explain their expectations about a paper prototype or a mockup.

Focus Group: A representative group of end users are requested to participate in an open discussion to analyze the graphical interface of a software product. In this method, participants are free to listen and talk to other group members. In this way, they can develop own ideas based on previous comments.

Checklist Verification: A usability specialist verifies if a graphical user interface meets a series of well-defined design specifications. A verification checklist helps inspectors to manage all of the details of usability that must be considered in a particular software product. This list

Pencil & Paper: The users evaluate aspects of a prototype on paper. They are free to modify the interface design with a pencil. Additionally, they can write their comments and make annotations to specify their observations in detail.

Perspective Based Usability Inspection: In each inspection session, the specialist focuses on a specific subset of usability issues covered by one of several usability perspectives. Each perspective provides the inspector a list of questions that represent the usability issues to check and a specific procedure for conducting the inspection. The assumption is that with focused attention and a well-defined procedure, each inspection session can detect a higher percentage of the problems related to the perspective used, and that the combination of different perspectives can uncover more problems than the combination of the same number of inspection sessions using a general inspection technique.

Field Observation / Field Study: This method involves a usability specialist observing user's natural behavior in their "natural habitat", the field where the daily activity takes place or the workplace where the software product will be implemented. The facilitator gives the user a task and observes, takes notes, and asks questions as the user employs the software product to complete the defined task. Observation can be direct, where the inspector is present during the task, or indirect, using special software to capture user actions on the computer and record the session.

Eye Tracking: This method involves measuring either where the user is looking (the point of gaze) or the motion of an eye during the use of a software product. There are several devices to perform this kind of evaluation such as: special monitors, specific cameras, sensors and even specialized software. By analyzing the visual path of the end users across the interface, it is possible to determine the relevant information, the sections that are ignored, the content which is overlooked any other gaze-related question.

Click Map / Scroll Map / Heat Map: Clickmaps shows where users click on a software interface. This information allows inspectors to identify the most popular sections, and see which sections users mistake for links. This map is often represented by colors which indicate the amount of clicks in a specific area. A click map can be obtained through the use of a special software tools.

Opinion Mining: This method refers to the use of natural language processing and text analysis to identify subjective information regarding the usability of a software product. For this purpose, a representative group of user have to write their opinion for certain software in three usability factors: effectiveness, efficiency and satisfaction. Then, these comments are analyzed using specialized techniques of Computer Science to determine how positive or negative are in each category [1].

Web Usability Evaluation Process: This method involves the decomposition of the usability concept into sub-characteristics and measurable attributes, which are then

associated with metrics in order to quantify them numerically. This technique has been specially developed for Web applications. The purpose is to provide feedback during all phases of the software development process. A complete model, including all the sub-characteristics attributes and their associated metrics, is provided by the authors of this method [3].

Retrospective Thinking Aloud: This method is another variant of user testing. It is a similar practice to the thinking aloud protocol, however in this method, users have to verbalize their thoughts after the user testing session activities, instead of during them. Users are requested to use the system and perform certain tasks in silence. Participants verbalize their thoughts afterward while they are watching a recording of their performance [20].

Cognitive Task Analysis: This techniques involves the process of learning about ordinary users by observing their interaction with a specific software product in order to understand in detail how they perform their tasks and achieve their intended goals. Tasks analysis helps to identify the tasks that a software application must support and can also help you refine the navigation or search. This method is focused on understanding tasks that require decision-making, problem-solving, memory, attention and judgment.

Usability Guidelines: A group of specialists have to evaluate the graphical interface of a software product according to pre-defined usability guidelines. Although this technique is similar to heuristic evaluation, the procedure is different. In this technique, each inspector can work individually. There is no need to rate the severity and criticality of each usability issue. The assessment tool is not necessarily a set of usability heuristics. Inspectors can even use guidelines that are provided by the software development company.

Card Sorting: This method can be used to verify the organization and structure of the information that appears in a software application. For this kind of evaluation, some paper cards are required. Each card has to contain a word or phrase written on one side. This expression has to represent a specific concept that is considered as part of the graphical user interface. Participants are given a stack of cards and are asked to group them together as it makes sense to them. They organize topics into categories and may also help to label these groups. If an accepted and standardized taxonomy becomes visible, it would be appropriate to apply that taxonomy in the interface.

Canvas Card Sorting: This technique is a variation of the classical card sorting. This method requires that users select the most valuable concepts and arrange them in a predefined template. In this version, the main categories are previously established, and users only have to place each card into one of the groups.

Retrospective Sense Making: This method is based on a retrospective protocol, in which users are asked to verbalize their thoughts after a set of tasks is completed. This specific technique establishes the use of open-ended questions in order to encourage users to process information from the long-term memory, providing justifications and explanations of certain actions they performed during their interaction with an interface. The questions should be oriented to analyze the cognitive process through which people experience problems and choose to perform certain actions, among alternative ones, in order to solve the problems experienced at a specific point in time [16].

Personas: This method involves the description of different fictitious users of the software application. These representations should include a brief profile of goals and characteristics that represent the needs of a larger group of real users. The evaluation involves an analysis of the graphical user interface considering the goals, possible behaviors, attitudes, motivations and business objectives of each profile.

User Workflow: This method establishes the elaboration of diagrams to represent all the paths that are available in a software system to perform a specific task. This diagram allows specialists to analyze the achievement of multiple goals which involve many sub-tasks. Additionally, it permits the examination of the different users' preferences and the order in which certain tasks are performed.

Cognitive Jogthrough: This method is an alternative version of the cognitive walkthrough. In this version, while inspectors are working through a series of tasks, they ask themselves a set of questions from the perspective of the user. The answers to these questions should be ranked according to the percentage of potential users are expected to have problems (from 0 to 3 in a Likert scale) [7].

Domain Specific Inspection: This method involves the use of a model [17] that can be adapted to any software domain. Specialists should determine the areas and attributes that are more relevant for software they are going to evaluate. The inspection should be performed according to the guidelines that have been established for each usability attribute.

Participatory Heuristic Evaluation: is an extension of the traditional heuristic evaluation where some principles are considered to evaluate the graphical user interface. Participatory heuristic evaluation uses the same technique. However, it involves the participation of end users in the evaluation process as "domain expert inspectors". Additionally, some additional heuristics are added to include some usability aspects that are not considered by the traditional Nielsen's proposal.

Semiotic Inspection Method: The purpose of this method is the analysis of the messages conveyed through the designer-to-user metacommunication. These messages are expressed with a broad range of signs and symbols in the interface, from one or more signification systems. The aim of the semiotic inspection method is the evaluation of these elements, searching for actual or potential problems of communication and redesign opportunities to improve the communication [6].

Usability & Communicability Evaluation Method: In this method, evaluators have to identify communication breakdowns while a representative amount of users interacts with the product software. There are thirteen expressions of communication breakdown or labels to categorize the problems of communicability and usability. The evaluator should interpret these issues and rebuild the message to identify possible improvements.

Simplified Pluralistic Walkthrough: Users and designers participate together in a meeting to evaluate new ideas regarding the graphical user interface of a software product. The method does not require a working prototype. They can develop a design from just ideas. The system designers can get valuable information about the users' tasks in addition to the comments on the design.

Simplified Streamlined Cognitive Walkthrough: This method establishes the same procedure than cognitive jogthough. The difference is that evaluators only required asking two questions at each step of the inspection [19]. Moreover, it involved to elaborate less documentation.

Music Performance Measurement Method: This method establishes that the usability of a product is measured by the extent to which specific users achieve specific goals in a specific environment. Some metrics are employed to determine qualitative data regarding usability. The technique indicates that the controlled experiments should be performed as close to a real work environment. A software tool called DRUM can be used to analyze log files.

4.2. Results by Software Domain

In this section, we detail the main domains of software applications that were reported in the primary studies. The results are summarized in Table 5.

Table 5. Main Software Domain Involved in the Studies

Software Domain	Number of software applications that were evaluated	Percentage
Health Informatics	50	20.33%
Education	36	14.63%
Software Development Tools	23	9.35%
E-Commerce	20	8.13%
Gaming	19	7.72%
Personal Information Manager	9	3.66%
Routes & System Guides	6	2.44%
General Purpose Systems	6	2.44%
Expert Systems	5	2.03%
Geographic Information Systems	5	2.03%
Software for Users' Communication	5	2.03%
Enterprise Resource Planning	4	1.63%
Tools & Utilities	4	1.63%
Other domains	39	15.85%
No information about the software	15	6.10%
TOTAL	**246**	**100.00%**

Most of the systems were related to *Health Informatics*. In this category were considered: apps for health, hospital management systems, clinical decision supports systems, electronic patient record systems, software for specialized medical devices, *etc*.

In *Education*, all kinds of software products that are used to support activities of learning and teaching were considered, such as E-Learning platforms, teaching tools and management systems for universities.

In the category of *Software Development Tools*, we reviewed all applications that are used to support any activity of the software development process. Some examples include IDEs, CASE tools, software to conduct test cases, *etc*.

In E-Commerce, we considered all software that is related to the sale of products and services. This category covered most of the transactional Web applications.

Personal Information Manager is related to the software that acquire, organize, maintain and retrieve personal information. Some examples include e-mail applications, software for managing credit cards, Web calendars, *etc*.

The category of *Routes & System Guides* includes all software that provides information about the available routes of buses and trains, including schedules and maps. In this domain, we also considered applications that guide users to specific locations according to real-time traffic information.

General Purpose Systems are related to Web application whose purpose is only to show information. For instance, the Web site of some companies shows information related to the products and services they offer, however, they do not provide mechanisms to buy online. Search engines were also considered in this category.

In *Expert Systems* were considered computer systems that emulate the decision-making ability of a human expert. The majority of these applications were related to medical expert systems that could diagnose a disease and give a prescription.

In *Geographic Information System*, we considered all applications that offer information about different places of a city based on the current location of the user. In the same way, the category of *Software for Users' Communication* includes mobile and Web applications that allow real-time communication.

In the category of *Enterprise Resource Planning*, software applications that are capable to collect, store, manage and interpret data from any business activity were considered.

Finally, the domain of *Tools & Utilities* includes applications that are provided by the operating system.

The category "other domains" includes: *Customer Relationship Management Systems*, *Human Resource Information Systems*, *Decision Support Systems*, *Software for Astronomy and Space Science*, *Financial Software*, *Project Management Tools*, *Social Networks*, *Banking Software Systems*, Real-*Time Location Systems*, *Operating Systems*, *Software for Musicians & Audio Processing*, *Software to Measure Psychological Aspects*, *Molecular Biology Software*, *Radio & Television Online Systems*, *Government & Public Sector Software Solutions*, *Software for Transport Sector*, *Software for Business Processes*, *Software for Industrial Factories*, *Software for Military, Security & Defense*, *Supply Chain Management Software*, *Software for Maritime Sector*, *Software for Aircraft*, *Software for Physics*, *Software for Biology*, *Photo Editors*, *Software Controlled by Brain* and *Mathematical & Statistical Software*.

Table 6. Main Usability Evaluation Methods by Software Domain

Software Domain	Method that was used to evaluate the usability	Number of times the method was used	Percentage (%)
Health Informatics	Survey / Questionnaire	28	30.43%
	User Testing - Thinking Aloud	16	17.39%
	User Testing	13	14.13%
	Heuristic Evaluation	10	10.87%
	Others	25	27.17%
Education	Survey / Questionnaire	19	28.79%
	User Testing	10	15.15%
	Interview	10	15.15%
	Heuristic Evaluation	7	10.61%
	Others	20	30.30%
Software Development Tools	Survey / Questionnaire	10	28.57%
	User Testing	7	20.00%
	Interview	5	14.29%
	User Testing – Thinking Aloud	4	11.43%
	Others	9	25.71%

In Table 6 and Table 7, we present the most commonly used methods for each domain. Although the differences, the survey is one of the most commonly used in all software domains.

Table 7. Main Usability Evaluation Methods by Software Domain

Software Domain	Method that was used to evaluate the usability	Number of times the method was used	Percentage (%)
E-Commerce	Heuristic Evaluation	9	25.00%
	Survey / Questionnaire	8	22.22%
	User Testing	5	13.89%
	Software Metrics	3	8.33%
	Others	11	30.56%
Gaming	Survey / Questionnaire	8	27.59%
	Heuristic Evaluation	6	20.69%
	User Testing	3	10.34%
	User Testing – Thinking Aloud	3	10.34%
	Others	9	31.03%
Personal Information Manager	Survey / Questionnaire	3	15.79%
	Heuristic Evaluation	3	15.79%
	User Testing	3	15.79%
	Software Metrics	3	15.79%
	Others	7	36.84%
Routes & System Guides	Survey / Questionnaire	3	25.00%
	Interview	2	16.67%
	Prototype Evaluation	2	16.67%
	Checklist Verification	2	16.67%
	Others	3	25.00%

General Purpose Systems	Survey / Questionnaire	4	36.36%
	User Testing	2	18.18%
	User Testing – Thinking Aloud	2	18.18%
	Others	3	27.27%
Expert Systems	Survey / Questionnaire	4	80.00%
	User Testing – Thinking Aloud	1	20.00%
	Others	0	00.00%
Geographic Information Systems	User Testing – Thinking Aloud	2	28.57%
	Others	5	71.43%
Software for Users' Communication	Heuristic Evaluation	2	25.00%
	User Testing	2	25.00%
	User Testing – Thinking Aloud	2	25.00%
	Others	2	25.00%
Enterprise Resource Planning	Survey / Questionnaire	2	22.22%
	Heuristic Evaluation	2	22.22%
	Software Metrics	2	22.22%
	Others	3	33.33%
Tools & Utilities	Survey / Questionnaire	2	28.57%
	Others	5	71.43%

4.3. Results by Type of Application

In order to determine the current trends by type of application, we developed a mapping (Figure 01) to represent the number of times that the most popular methods were used in these particular contexts. It possible to determine that the trend, that was established in Section 4, continues. Questionnaire is the most used technique because of its technical simplicity. There is no significant difference between user testing and heuristic evaluation. However, "heuristic evaluation" is more reported than "user testing" for the context of mobile applications. Finally, thinking aloud protocol is the less employed for these contexts except mobiles, where the interview scored less.

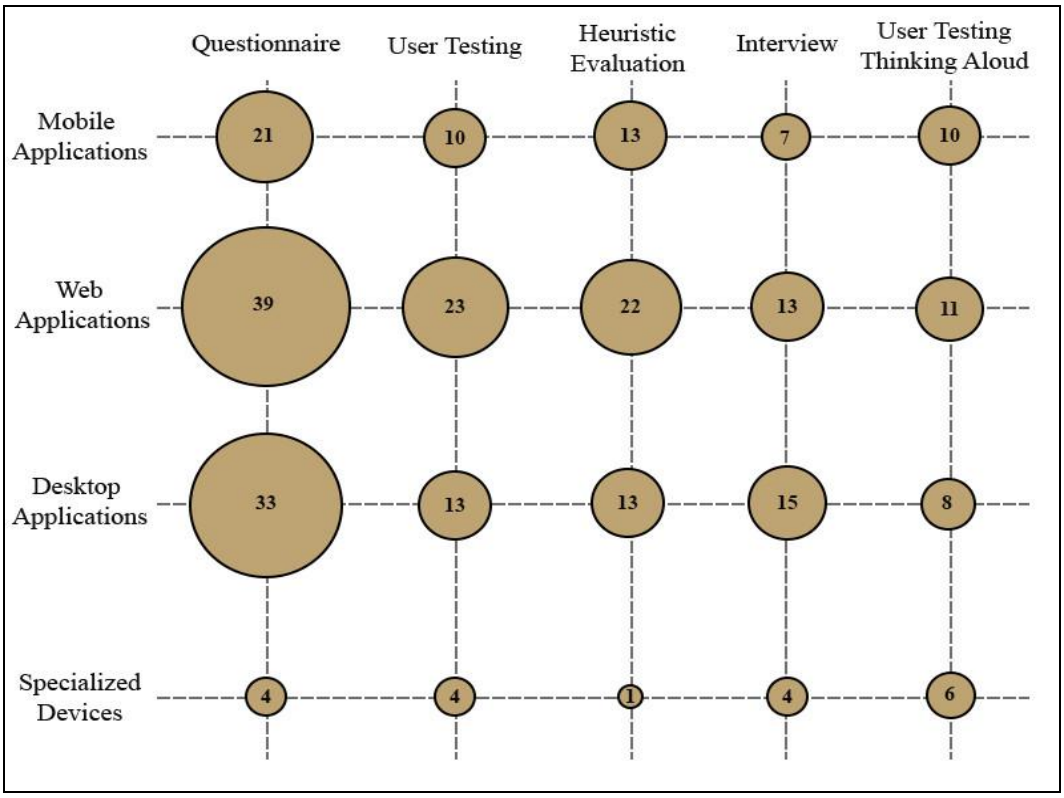

Figure 1. Categorical Bubble Plot About of the Main Usability Evaluation Methods by Type of Application

5. Conclusions and Future Works

Several evaluations methods have been proposed to determine the level of usability of software applications. Despite this large amount of methods, there is still to determine the most suitable technique for a particular scenario. In this study, we performed a systematic mapping review of the use of usability evaluations methods for software development processes. Our study is intended to serve as contribution to support decision making in the choice of a technique.

Following a predefined protocol, we identified 1169 studies, from which 215 were selected. This work allowed to determine that: (1) questionnaire, (2) user testing, (3) heuristic evaluation, (4) interview, and (5) thinking aloud protocol are the most employed techniques according to the literature. Additionally, in this study we have determined the types of applications that are frequently reported in the literature as part of a usability evaluation in software developments. Most of the application belong to the categories of (1) Health Informatics, (2) Education, (3) Software Development, (4) E-Commerce and (5) Gaming.

From the analysis, we further note that some techniques are adapted to cover all aspects of usability in some types of software products. The emergence of hybrid categories has forced scholars to propose particular assessment tools such as usability questionnaires for specific domains, heuristics for a particular kind of software, variants of a usability method, *etc*. There is also a need for a more in-depth analysis in each category, especially in the methodology, establishing the differences and how these affect the final result.

Appendix A. Papers Identified in the Systematic Mapping Study

This information is available at:

http://inform.pucp.edu.pe/~jpowsang/usability/mapping_study_appendix.htm

References

[1] A. M. El-Halees, "Software Usability Evaluation Using Opinion Mining", Journal of Software, vol. 9, no. 2, **(2014)**, pp. 343-349.
[2] A. Fernandez, E. Insfran and S. Abrahão, "Usability Evaluation Methods for the Web: A Systematic Mapping Study", Information and Software Technology, vol. 53, no. 8, **(2011)**, pp. 789-817.
[3] A. Fernandez, S. Abrahão and E. Insfran, "A Web Usability Evaluation Process for Model-Driven Web Development", Proceedings of the 23rd International Conference on Advanced Information Systems Engineering, London, United Kingdom, (**2011**) June 20-24.
[4] A. Holzinger, "Usability Engineering Methods for Software Developers", Communications ACM, vol. 48, no. 1, **(2005)**, pp.71-74.
[5] B. Kitchenham and S. Charters, "Guidelines for Performing Systematic Literature Reviews in Software Engineering", Keele University and Durham University Joint Report, Tech. Rep., **(2007)**.
[6] C. S. de Souza, C. F. Leitão, R. O. Prates and E. J. da Silva, "The Semiotic Inspection Method", Proceedings of VII Brazilian symposium on Human factors in computing systems, Natal, Brazil, **(2006)** November 19-22.
[7] D. E. Rowley and D. G. Rhoades, "The Cognitive Jogthrough", Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, California, USA, **(1992)** May 3-7.
[8] F. Gündüz and A. K. Pathan, "On the Key Factors of Usability in Small-sized Mobile Touch-Screen Application", International Journal of Multimedia and Ubiquitous Engineering, vol. 8, no.3, **(2013)**, pp. 115-138.
[9] F. Paz and J. A. Pow-Sang, "Current Trends in Usability Evaluation Methods: A Systematic Review", Proceedings of the 8th International Conference on Advanced Software Engineering & Its Applications, Hainan, China, **(2014)** December 20-23.
[10] F. Paz and J. A. Pow-Sang, "Usability Evaluation Methods for Software Development: A Systematic Mapping Study", Proceedings of the 8th International Conference on Advanced Software Engineering & Its Applications, Jeju Island, Korea, **(2015)** November 25-28.
[11] F. Paz, D. Villanueva, C. Rusu, S. Roncagliolo and J. A. Pow-Sang, "Experimental Evaluation of Usability Heuristics", Proceedings of the Tenth International Conference on Information Technology: New Generations, Las Vegas, NV, USA, **(2013)** April 15-17.

[12] ISO, "Ergonomic Requirements for Office Work with Visual Display Terminals (vdts), -- Part 11: Guidance of Usability", International Organization for Standardization, Geneva, Switzerland, ISO 9241-11:1998, **(1998)**.
[13] ISO, "Software Engineering – Product Quality – Part 1: Quality Model", International Organization for Standardization, Geneva, Switzerland, ISO 9126-1:2001, **(2001)**.
[14] J. Nielsen, "Usability Engineering", Academic Press, San Diego, CA, USA, **(1993)**.
[15] N. Vatankhah, K. T. Wei and S. Letchmunan, "Usability Measurement of Malaysian Online Tourism Websites", International Journal of Software Engineering and Its Applications, vol. 8, no. 12, **(2014)**, pp. 1-18.
[16] P. Balatsoukas, J. Ainsworth, R. Williams, E. Carruthers, C. Davies, J. McGrath, A. Akbarov, C. Soiland-Reyes, S. Badiyani and I. Buchan, "Verbal Protocols for Assessing the Usability of Clinical Decision Support: The Retrospective Sense Making Protocol", Technology and Informatics, vol. 192, **(2013)**, pp. 283-287.
[17] R. AlRoobaea, A. H. Al-Badi and P. J. Mayhew, "Generating a Domain Specific Inspection Evaluation Method through an Adaptive Framework", International Journal of Advanced Computer Science and Applications, vol. 4, no. 6, **(2013)**, pp. 72-91.
[18] R. Otaiza, "Evaluating the Usability of Transactional Web Sites", Third International Conference on Advances in Computer-Human Interactions, Saint Maarten, **(2010)** February 10-15.
[19] R. Spencer, "The Streamlined Cognitive Walkthrough Method, Working Around Social Constraints Encountered in a Software Development Company", Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, California, USA, **(1992)** May 3-7.
[20] S. Elling, L. Lentz and M. de Jong, "Retrospective think-aloud method: using eye movements as an extra cue for participants' verbalizations", Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Vancouver, Canada, **(2011)** May 7-12.
[21] Z. Zhang, V. Basili and B. Shneiderman, "Perspective-based Usability Inspection: An Empirical Validation of Efficacy", Empirical Software Engineering, vol. 4, **(1999)**, 43-69.

Authors

Freddy Paz is part-time professor at *Pontificia Universidad Católica del Perú* (PUCP). His research interests include human-computer interaction and programming languages. He received his Master Degree in Informatics and Informatics Engineering from PUCP - Peru, and *Pontificia Universidad Católica de Valparaíso* (PUCV) - Chile. He is currently a doctoral student in Engineering. He has a BSc degree in System Engineering from *Universidad Nacional Pedro Ruiz Gallo* (UNPRG), Peru.

José Antonio Pow-Sang is full professor and executive director of the Postgraduate School at *Pontificia Universidad Católica del Perú* (PUCP). His research interests include empirical software engineering, software metrics, software engineering education, and human-computer interaction. He is Ph.D. in Informatics Engineering, and he has a Master in Software Engineering from Technical University of Madrid, Spain. He has a BSc degree in Informatics Engineering from PUCP. He is a senior member of IEEE Computer Society and a member of ACM.

