Automated Tools for Usability Evaluation: A Systematic Mapping Study

John W. Castro[1(✉)], Ignacio Garnica[1], and Luis A. Rojas[2]

[1] Departamento de Ingeniería Informática y Ciencias de la Computación, Universidad de Atacama, Copiapó, Chile
john.castro@uda.cl, ignacio.garnica.14@alumnos.uda.cl

[2] Facultad de Ciencias Empresariales, Departamento de Ciencias de la Computación y Tecnologías de la Información, Universidad del Bío-Bío, Chillán, Chile
lrojas.larp@gmail.com

Abstract. Usability is one of the most critical indicators in determining the quality of a software product. It corresponds to how users can use a software system to achieve specific objectives with effectiveness, efficiency, and satisfaction. A usability evaluation is necessary to ensure that the software system is usable, but this has certain disadvantages (e.g., a high cost of time and budget for the evaluation to be implemented). While these disadvantages can be a bit daunting despite the benefits they provide, some tools can automatically generate and support usability testing. We conducted a systematic mapping study to identify the tools that support automatic usability evaluation. We identified a total of 15 primary studies. In addition, we classify the tools into four categories: measure usability, support usability evaluation, detect usability problems, and correct usability problems. We identified that the automatic evaluation of the usability of web platforms and mobile devices is the most interesting.

Keywords: Usability · Evaluation · Tool · Automation

1 Introduction

Currently, there is a growth of developed software systems, causing an increased demand for higher quality systems, which can be ensured with specific standardized measures and methods through different activities and techniques. One of the essential measures when developing a software system is usability [1]. Usability is the extent to which users use a system, product, or service effectively and with satisfaction, given a context of use [2]. Usability is also related to the acceptability, by users, of a specific system, considering that it is good enough to meet the needs of users [1]. To ensure that these requirements are met, the developed systems must undergo a usability evaluation [3, 4].

Despite the importance of usability evaluation for any software system, it has certain disadvantages, such as a high cost of time and budget given its characteristics. Additionally, some techniques related to usability evaluation need at least one usability expert to be implemented [3, 4]. Guidelines, metrics, and heuristics can guide this expert to

G. Meiselwitz (Ed.): HCII 2022, LNCS 13315, pp. 28–46, 2022.
https://doi.org/10.1007/978-3-031-05061-9_3

support the work of usability evaluation. However, this expert evaluator will always provide a certain level of subjectivity in their analysis [4, 5]. Although these disadvantages can be discouraging, despite the benefits they provide considering the finished software product, they can be mitigated by implementing usability evaluation tools [5–8].

Usability evaluation tools are systems that support this task. Many tools directly benefit usability evaluation activities in an automated way, allowing, for example, during a usability test to store user registration data such as (i) keystrokes, (ii) clicks made with the mouse, and (iii) the distances traveled by the mouse pointer, among others. These tools allow, in some cases, to analyze the data collected to provide feedback to developers and usability experts, providing information on usability errors and, depending on the tool, automatically correcting them [5–9].

Currently, there is a wide variety of these tools. However, the related literature is composed of a set of independent publications. To the best of our knowledge, no study has comprehensively focused on this literature nor reports on the current state of automated tools for usability evaluation. This research seeks to generate a body of knowledge to classify the tools that support the automatic evaluation of usability. For this, we conducted a systematic mapping study (SMS).

This paper is organized as follows. In Sect. 2, we present the related work. In Sect. 3, we describe the research method of the SMS. In Sect. 4, we discuss the results of the SMS. Section 5 presents possible threats to validity, and finally, the conclusions are presented in Sect. 6.

2 Related Work

From our pilot search, we found that there were only four [4, 10–12] literature reviews related to our research. The first paper by Ivory and Hearst [4] reported the state-of-the-art usability evaluation methods, organized according to a taxonomy that emphasizes the role of automation. Ivory and Hearst [4] focused their efforts on identifying aspects of usability evaluation automation that are useful in future research and suggested new ways to expand existing approaches to better support usability evaluation. This study is interpreted as a precursor to automated approaches that, over time, became the development of tools that allow automatic evaluation of usability. Throughout his study, several tools are named, although not as sophisticated as those that currently exist, considering the year of publication of this study.

The second paper [10] reported *widgets* to help testers in the early evaluation of user interfaces. The authors explain that these *widgets* can detect certain ergonomic inconsistencies in the design of user interfaces. This study does not perform an SMS, and it focuses on exposing the *widgets* that were known. The authors explain the *widgets* in terms of functionality and application and show their experimental phase where they are tested. This study shows the *widgets* in a period before the one we consider (i.e., between 2016 and 2021), so the study is not considered in our research work.

The third paper by Bakaev et al. [11] provided an overview of the methods and tools of traditional, semi-automated, and automated approaches to website usability evaluation. The main difference from our research work, apart from the fact that the authors do not perform an SMS, is that Bakaev et al. [11] focused only on tools that support automated

usability evaluation of web user interfaces. In contrast, we focus our efforts on knowing the current panorama of these tools, whether they are focused on the web and desktop applications and mobile devices. Furthermore, the work of Bakaev et al. [11] only briefly describe the tools.

Finally, Khasnis et al. [12] presented a series of tools that support the usability evaluation in their research work, briefly explaining their operation, advantages, and disadvantages. It is important to note that the authors do not perform an SMS, as in this study. Furthermore, Khasnis et al. [12] focused on relating automatic usability evaluation tools with usability evaluation methods. Our approach focuses on relating the reported tools to the catalog of usability evaluation techniques proposed by [13, 14].

After analyzing these papers, we find that the SMS reported in this paper differs from the above reviews in that it aims not only to identify the automated tools to support the usability evaluation but also to (i) identify the techniques related to evaluation that benefit from these tools, (ii) determine the existing problems and challenges of using automated tools for usability evaluation and (iii) classify these tools. None of the reviews in the literature address this issue. Therefore, it is necessary to investigate the current state of automated tools for usability evaluation.

3 Research Method

The secondary study presented in this paper has been developed following the guidelines established by Kitchenham et al. [15] for conducting an SMS. Following these guidelines, the activities we carried out were: (i) formulating the research questions, (ii) defining the search strategies, (iii) selecting the primary studies, (iv) extracting the data, and (v) synthesizing the extracted data.

3.1 Research Questions

The information extracted from the primary studies aims to answer the following research questions: (RQ1) What are the automated tools that support the usability evaluation? (RQ2) Which usability evaluation-related techniques benefit from automated tools? (RQ3) What are the existing problems and challenges of using automated tools for usability evaluation? (RQ4) How can automated tools for usability evaluation be classified?

3.2 Define the Search Strategy

The SMS begins with identifying the keywords, for which it is necessary to find an initial set of articles that answer the research questions. This set is known as the Control Group (CG). The CG is a set of research papers representing, as accurately as possible, the set of primary studies that answers the research questions of the SMS [16]. Furthermore, the CG serves as a source of training samples for refining search strings and determining the sensitivity of the search strategy defined for the SMS. Keep in mind that a highly sensitive search strategy will retrieve many results. However, many of these may be unwanted articles, and a more precise search strategy will retrieve a few articles. However, it may

miss many studies that may be useful for research. Therefore, the formation of a CG must have a balance between these two factors [16]. To form the CG, a traditional search for studies related to the research context and, according to the previous explanation, that responds to the research questions was carried out. As a result of this search, six studies were identified [5, 7, 8, 17–19]. Before building the search string, it is verified if the CG studies are found in the Scopus database since it is the one that hosts the most studies. Within Scopus, there are five of the six that belong to the CG; they are [5, 7, 8, 18, 19]. Therefore, we can ensure that Scopus is the best option for research.

To obtain the keywords, a table was generated with the frequency of all the words and combinations of words that appeared in the CG articles, with the help of the Atlas.ti 9 software [20]. We selected only those words directly related to the research questions and that were present in a significant percentage of the CG articles. Subsequently, each one of the words obtained was assigned a value from 0 to 1, determined by its frequency of use, so that the word most frequently repeated in the various CG articles had the value 1. Table 1 shows a fragment of the list of words obtained as a result of this selection process. It shows the words, the percentage of CG studies it appeared in (coverage), the frequency of its appearance throughout the CG studies, and its assigned weight, based on the two previous columns. The weight is calculated based on the percentage of appearance and the frequency as follows (see Eq. 1):

$$((Word\ \ coverage)/(Maximum\ \ coverage) + (Word\ \ frequency)/(Maximum\ \ frequency))/2 \quad (1)$$

Table 1. Fragment of the list of words obtained from the selection process.

Words	Coverage (%)	Frequency	Weight
Usability	100	1156	1
Evaluation	100	577	0.7496
User	100	388	0.6678
Tool	100	240	0.6038
Interface	100	147	0.5636

3.3 Formation of the Search String

Once the keywords were identified, several search strings were constructed. For constructing the strings, four components are considered that correspond to a classification of the words considered. To define the components, the context of this research was considered, that is, knowing the current panorama of automatic tools that allow the usability evaluation to be supported. The defined components were the following: (i) tools, (ii) automation, (iii) evaluation, and (iv) usability. The logical operator AND was used to join

each of these components, while the logical operator OR was used to include synonyms of words from the same component. A total of four search strings were constructed, as shown in Table 2. We used these strings to search for CG studies within the Scopus database. It is important to remember that five of the six CG studies are in the Scopus database.

Table 2. Search strings.

ID	Search string	Studies found	GC found	Ratio X	Ratio Y	Average
1	(usability OR "user experience") AND (evaluation OR testing OR measure OR evaluating OR study OR evaluate OR tests OR assess) AND (tool OR systems OR applications OR tools OR software OR system OR application OR product) AND (automated OR automatic OR automatically OR automating)	2620	5	0.8333	0.0019	0.4176
2	(usability) AND (evaluation OR testing OR measure) AND (tool OR systems OR applications) AND (automated OR automatic OR automatically)	1004	5	0.8333	0.0049	0.4191
3	(usability OR "user experience") AND (evaluation OR testing) AND (tool OR tools OR software OR systems) AND (automated OR automatic)	912	5	0.8333	0.0054	0.4194
4	usability AND (evaluation OR testing OR evaluate OR study) AND (tool OR software OR systems) AND (automated OR automatic)	1304	5	0.8333	0.0038	0.4185

Table 2 shows the number of studies found and the number of CG articles found for each search string tested. All search strings find all five GC studies. Because of this, it was necessary to use additional indicators. These indicators are the X ratio (see Eq. 2), the Y ratio (see Eq. 3), and the average between both (see Eq. 4).

$$XRatio = (No.\ of\ articles\ found\ in\ the\ control\ group) / (Total\ of\ articles\ in\ the\ control\ group) \quad (2)$$

$$YRatio = (No.\ of\ articles\ found\ from\ the\ control\ group) / (Total\ of\ articles\ found\ per\ search\ string) \quad (3)$$

$$Average = (XRatio + YRatio)/2 \quad (4)$$

As shown in Table 2, the X ratio remains the same for all search strings. This is because, with all strings tested in the Scopus database, the same number of articles belonging to the CG was found. However, the Y ratio shows specific differences since it is based on calculating the proportion of the CG articles found in the total of the results obtained by each string. The string with the highest Y ratio is string 3. To ensure that the selected string is the ideal one for our investigation, the average between the X ratio and the Y ratio is calculated. According to Table 2, string 3 has the highest average, so it is selected as the best search string. The structure of the final search string is shown in Table 3.

Table 3. Final search string.

Keywords						
usability OR "user experience"	AND	evaluation OR testing	AND	tool OR tools OR software OR system	AND	automated OR automatic

Although the search string tests were performed in Scopus, the largest database of peer-reviewed literature [21], the searches were also performed in the IEEE Xplore and Web of Science (WoS) in order to acquire more results. In the search, only studies from 2016 to September 2021 are considered. The databases were analyzed sequentially, using the search fields shown in Table 4. The search fields used were determined by the options provided by each database, due to the different query syntaxes [22–24]. If a duplicate appeared, the first result was kept.

Table 4. Search field per database.

Database	Search fields	Number of results
Scopus	"Title OR Abstract OR Keywords"	904
IEEE Xplore	"Abstract"	162
Web of Science	"Title OR Abstract OR Keywords"	191

3.4 Inclusion and Exclusion Criteria

The inclusion criteria used to select the primary studies are summarized below:

- The article describes one or several tools that support the evaluation of usability or user experience, explaining in detail its operation (e.g., implemented algorithms, architecture, methodologies, theory involved).
- The article reports a testing phase in actual use cases where the tools are tested, and conclusive results are reported, demonstrating that the described tool meets the objective of supporting the evaluation of usability.

It is essential to mention that selecting a study must meet both inclusion criteria. In contrast to this, the exclusion criteria are as follows:

- The tools reported in the study do not perform or support automatic usability evaluation.
- The article does not explain the operation of the presented tools in detail.
- The article does not report a testing phase of the tools.
- The testing phase reported in the article does not deliver conclusive results that answer the research questions.
- The results of the testing phase reported in the article do not show that the tools described meet the objective of supporting usability evaluation automatically.
- The tools described in the article deliver only raw data without any analysis or critique.
- The tool presented in the article is a framework.
- The article is written in a language other than English.

Note that it is enough for a study to meet one of the exclusion criteria to be discarded.

3.5 Select the Studies

A total of 1811 papers were found in the different databases. After excluding duplicate articles, the number was reduced to 1257. Subsequently, a selection of studies was made by applying the inclusion and exclusion criteria to the title and abstract of each of these 1257 studies. The selected articles were validated during a consensus meeting, in which we analyzed the abstracts of articles with conflicting decisions, thus reducing the total to 133 pre-selected articles. After the meeting, the selection criteria were again applied to the full text of the remaining articles. Figure 1 shows the entire filtering and analysis

process with the inclusion and exclusion criteria used to select 15 papers. A complete list of the primary studies can be found in Appendix A. The results of applying the different filters during the selection process for each database can be seen in Table 5.

Table 5. Number of remaining studies after filtering the database results.

Database	Studies found	Duplicate-free	Pre-selected studies	Primary studies
Scopus	912	904	110	13
IEEE Xplore	306	162	16	2
Web of Science	593	191	7	0
TOTAL	**1811**	**1257**	**133**	**15**

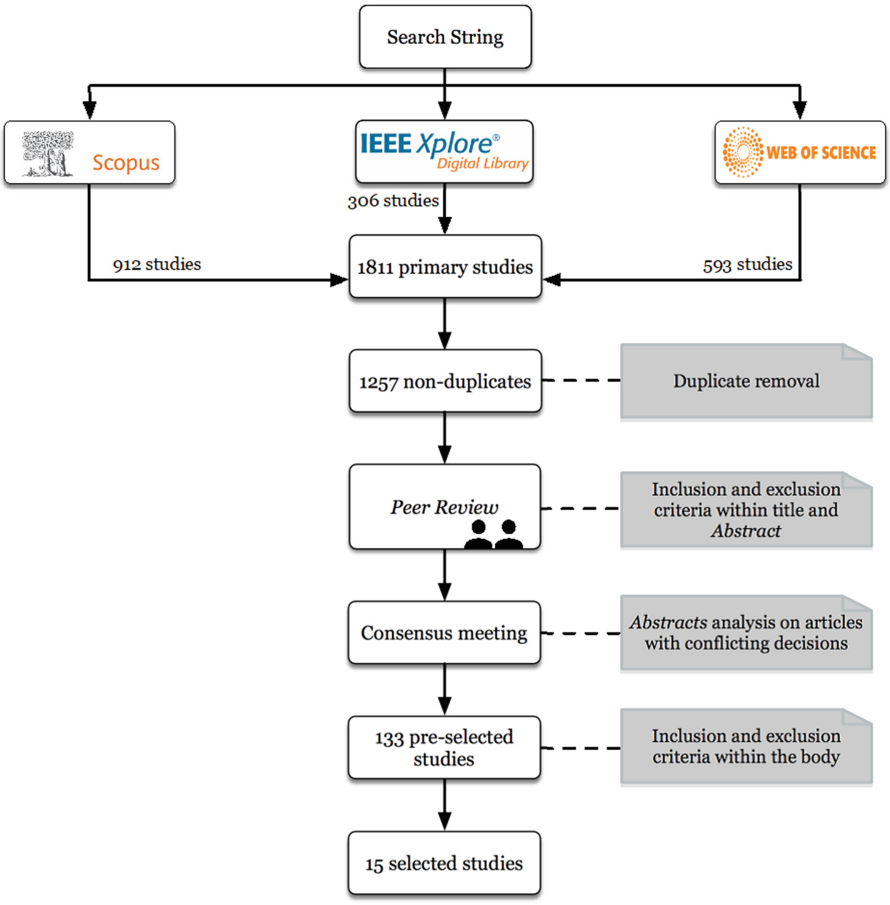

Fig. 1. Steps followed during the systematic mapping study.

4 Results and Discussion

Figure 2 synthesizes the results using two bubble scatter plots. The upper graph represents the number of articles published per year, according to publication type (journal, book chapter, or conference). Similarly, the lower graph plots the publication type against the classification tools (see Sect. 4.4). Thus, the bubbles are located at the intersections between the two axes and their size is proportional to the number of publications for each combination of values.

As can be seen in the upper part of Fig. 2, in 2016, only two studies were found. An excellent interest in tools that support the usability evaluation can be seen in 2017, where five of the 15 primary studies are concentrated. This interest progressively declines, finding three studies in 2018, two in 2019, and only one in 2020. Interest in this area of research recovers a little in 2021, with two studies.

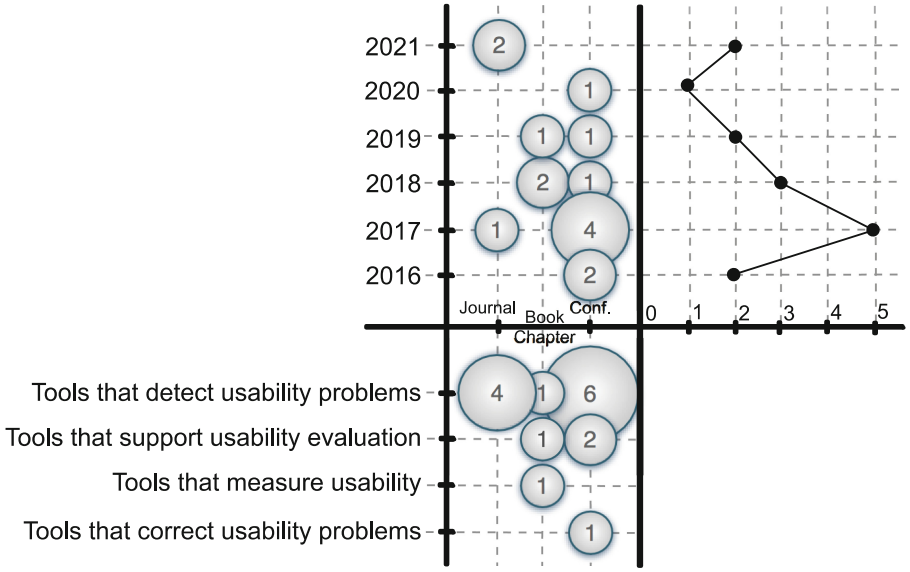

Fig. 2. Mapping for the primary study distribution between the classification of the tools along with the type of publication.

In the lower part of Fig. 2, it is seen that the classification that includes the most tools is "Tools that detect usability problems," followed by "Tools that support usability evaluation." Next, each research question will be answered.

4.1 Automated Tools to Support Usability Evaluation

In this section, we answer the first research question: *What are the automated tools that support the usability evaluation?* From the analysis of the 15 primary studies, 14 tools are obtained, which will be described below.

- **MOBILICS** [PS1] is an extension of USABILICS, so it inherits its methodology. This extension arises from the need to evaluate the usability of web pages in mobile environments considering the touch elements of these devices. MOBILICS performs the usability evaluation by comparing the actual interaction of a user performing a usability test with the interaction predefined by the evaluator who designs the test.
- **Environment for Supporting Interactive Systems Evaluation** [PS2] is a tool that automatically supports the usability evaluation of desktop web user interfaces. This tool performs usability evaluation by detecting usability problems through indicators, using usability data obtained from objective (e.g., an ergonomic guideline inspector) and subjective (through questionnaires) methods.
- **USF (Usability Smell Finder)** [PS3] is a tool that automatically supports usability evaluation of web user interfaces in desktop environments, operating as Software-as-a-Service (SaaS). This tool performs usability evaluation focusing on detecting usability smells, which serve as clues that point to possible usability problems.
- **MUSE (Mobile Usability Smell Evaluation)** [PS4] supports automatic usability evaluation of web user interfaces in desktop and mobile environments. MUSE records user interaction in usability testing sessions. Thanks to its proxy server approach, it can inject JavaScript code to the website to be evaluated without the need for the owner to do so manually.
- **Kobold** [PS5] supports automatic usability testing of web user interfaces in desktop environments, running as SaaS. This tool performs a usability evaluation focusing on detecting usability smells, providing refactorings that can be implemented manually, semi-automatically, or automatically to correct usability problems. Kobold is built on USF, so it uses a similar strategy when detecting usability smells.
- **Plain** [PS6] supports automatic usability evaluation of mobile applications. Plain is an Eclipse Plug-in that allows predicting the usability of a user interface by comparing usability metrics (e.g., composition, symmetry) with the properties of the elements that make up the mobile user interface to be evaluated.
- **UTAssistant** [PS7–PS9] allows supporting the usability evaluation automatically of user interfaces with a web focus. UTAssistant is a web platform that supports usability evaluation work by collecting mouse and keyboard log data during usability testing and allowing audio and video recording (both screen and user face).
- **Guideliner** [PS10] supports automatic usability evaluation of web user interfaces, both desktop, and mobile environments. Guideliner comprises several Java modules and uses Selenium WebDriver as its usability evaluation engine, allowing to search and analyze user interface elements and their features, comparing their values to guidelines to determine usability issues.
- **I2Evaluator** [PS11] supports automatic usability evaluation of web user interfaces, both mobile and desktop environments. i2evaluator seeks to measure user interfaces using aesthetic metrics (e.g., the balance of user interface objects) by incorporating an image decomposition algorithm that helps detect user interface elements to perform metric calculations.

- **PlatoS** [PS12] supports automatic usability evaluation of user interfaces in mobile application environments. The evaluator must create the tasks to perform in the usability tests and simulate the ideal interaction with the user interface. Using predefined usability metrics, PlatoS performs a statistical analysis of the times and actions performed by the evaluator and users to detect usability problems.
- **OwlEye** [PS13] supports automatic usability evaluation of mobile application user interfaces. OwlEye implements a CNN (Convolutional Neural Network) model for usability problem detection. With a set of 66,000 screenshots of more than 9,300 Android applications and using the CNN model, OwlEye can detect problems in user interfaces with a high level of efficiency.
- **ADUE (Automatic Domain Usability Evaluation)** [PS14] automatically allows usability evaluation of desktop applications. ADUE detects domain usability issues based on the domain usability approach. This approach covers aspects related more to the content of the elements that make up the user interface than to their characteristics. ADUE shows the tester the errors and associated components and provides recommendations to correct these problems.
- **GTmetrix** [PS15] supports automatic usability testing of web pages in desktop environments, detecting performance issues by comparing page metrics against 23 rules (related to performance aspects) provided by Yahoo. These values are compared with those detected on the page, and with this the problems to be solved are determined.
- **Dareboost** [PS15] supports the automatic usability evaluation of web pages in mobile environments, using performance metrics (e.g., load times) and comparing them with the metrics obtained from the elements of the web page to be analyzed. The tool provides reports showing the general score of the page, the number of problems, and the improvements recommended for their respective corrections.

4.2 Usability Evaluation Techniques Benefited by Automated Tools

This section answers the second research question: *Which usability evaluation-related techniques benefit from automated tools?* It is essential to mention that the tools have different functionalities and cover usability evaluation differently; therefore, the techniques benefited using these vary according to each case. The techniques used are described below.

- **Interaction Logging** is a technique that records the complete interaction of a user testing a system in such a way that it can be fully reproduced in real-time [25]. The tools that support this technique are Environment for Supporting Interactive Systems Evaluation [PS2], USF [PS3], MUSE [PS4], Kobold [PS5], UTAssistant [PS7–PS9], and PlatoS [PS12].
- **Standards Conformance Inspection** is an inspection method where technology specialists inspect the system determining whether it meets the previously proposed standards [25]. Tools that support this technique are Plain [PS6], I2Evaluator [PS11], PlatoS [PS12], and GTmetrix [PS15].
- **Questionnaires** are an indirect method for studying the user interface that allows knowing the user's opinions about the use of the interface but not giving direct information about it [1]. This technique is supported by two tools: Environment for Supporting Interactive Systems Evaluation [PS2] and UTAssistant [PS7–PS9].

- **Consistency Inspection** is a technique in which a team of designers inspects a set of interfaces for a family of products [25]. This technique is supported by two tools: Environment for Supporting Interactive Systems Evaluation [PS2] and ADUE [PS14].
- **Guidelines review** is a technique in which experts check the conformity of the user interface with the organizational guidelines document or with other guidelines [26]. This technique is supported by Guideliner [PS10] and GTmetrix [PS15].
- **Continuous Recording of User Performance** is a technique that emerges from the evaluation during the active use of the software that is intended to be evaluated [26]. The software architecture should make it easy for system administrators to collect data about system usage patterns, user performance speed, error rate, or frequency of online help replays. This technique is supported by the Environment for Supporting Interactive Systems Evaluation [PS2] and MUSE [PS4] tools.
- **Usage Logging** is a technique that seeks to record the user's actual usage in their interaction with a system, which implies having the computer automatically collect statistics about the detailed usage of the system [1]. This technique is supported by a single tool: MOBILICS [PS1].
- **Video/Audio Recording**, s its name suggests, seeks to generate audiovisual records of user interaction with systems in usability tests [27]. This technique is supported by a single tool: UTAssistant [PS7–PS9].
- **Time Keystroke Logging** is a technique that seeks to generate a record of each keystroke pressed by a user testing a system [25]. Each of these keystrokes is stored along with the exact time the event occurred. This technique is supported by a single tool: PlatoS [PS12].
- **Performance Metrics** is a technique in which essential aspects of the actual use of the software system to be evaluated are quantified, either in a controlled laboratory environment or in the usual work environment [28]. This technique is supported by a single tool: Dareboost [PS15].
- **Heuristic Evaluation** is a technique in which a usability expert observes an interface and tries to obtain an opinion on its good and bad characteristics [1]. This heuristic evaluation technique is supported by a single tool: OwlEye [PS13].

4.3 Problems and Challenges of Using Automated Tools

This section answers the third research question: *What are the existing problems and challenges of using automated tools for usability evaluation?* The main problems and challenges identified in the primary studies are described below.

- **Event Detection in Software Systems** is a technical problem based on the difficulties reported by the authors to identify when events occur in the systems to be evaluated. Goncalves et al. [PS1] reported that considering the MOBILICS tool, the main challenge was detecting events related to touch screens (i.e., *touchstar*, *touchmove* and *touchend*).
- **Detection of Indicators and Thresholds** can be considered an implementation difficulty when determining the metrics to use and how to capture the data that will make the corresponding comparisons with the user interface elements to be evaluated. Assilla et al. [PS2] referred to this problem in the context of the presentation of the Environment for Supporting Interactive Systems Evaluation tool.

- **Validation of Metrics** corresponds to the difficulty of choosing the metrics and quality standards so that the results of detecting usability problems are accurate. Generally, the tools that are based on these indicators must consider analysis models that allow detecting aspects of the user interface and translating them into values that can be interpreted and compared with these metrics. This is precisely the problem presented by the I2Evaluator tool [PS11].
- **A usability expert is still needed in some cases.** This is the main problem that tools seek to automate the usability evaluation. During their development, it must be determined how these tools will guarantee results that allow delivering a synthesis that defines the usability problems of a user interface. Avoiding depending on usability experts is one of the general problems [PS3].
- **General limitations and tool performance improvement.** It corresponds to the challenges reported in the primary studies [PS5, PS6, PS10]. Grigera et al. [PS5] considered improving the accuracy in detecting usability issues to automate new refactorings and select the most suitable one. Soui et al. [PS6] stated that some issues related to quality defect detection need to be investigated. In this way, the authors plan some refactoring operations (e.g., reorganization of the content of the mobile user interface). Marenkov et al. [PS10] specified the tool's limitations considering that it focuses on web environments, indicating that web pages that use Flash or Java Applets are not considered for the use of Guideliner.
- **Tools cannot completely replace manual evaluation.** The use of automated tools has several advantages, such as suitability for large-scale evaluation and less effort in terms of time. However, it is considered essential that these tools cannot completely replace manual tests since usability problems can be found but not how serious the problem is. Therefore, it is necessary to have a specific criterion that the evaluator must exercise based on the interpretation he wants to give to the information provided by the tool [PS15].

4.4 Classification of Automated Tools

In this section, the last research question is answered: *How can automated tools for usability evaluation be classified?* After analyzing the primary studies and the functionalities of each tool reported in each study, a total of four categories were identified, which will be described below.

- **Tools that measure usability.** The tools that belong to this category perform analysis of software systems and deliver an indicator (e.g., percentage of usability, a rating from 1 to 10) that describes the system's usability. These tools do not provide a very detailed analysis, nor do they provide feedback on the specific errors of the analyzed system in terms of usability. In this category, there is only the I2Evaluator tool [PS11].
- **Tools that support usability evaluation.** Tools belonging to this category perform analysis of software systems, provide an indicator that describes the usability of a system, and provide valuable functions that support the usability evaluation. Among these additional functions are (i) automated data capture (e.g., event log, log files), (ii) generation of forms for usability surveys, and (iii) timelines that support evaluation traceability usability, among others. In this category, there is only UTAssistant [PS7–PS9].

- **Tools that detect usability problems.** The tools that belong to this category perform an analysis of software systems and provide feedback on the specific errors found related to usability. The tool displays these errors, for example, in the form of alerts, reports, warnings. To this category belong the tools MOBILICS [PS1], Environment for Supporting Interactive Systems Evaluation [PS2], USF [PS3], MUSE [PS4], Plain [PS6], Guideliner [PS10], PlatoS [PS12], OwlEye [PS13], ADUE [PS14], GTmetrix [PS15] and Dareboost [PS15].
- **Tools that correct usability problems.** The tools that belong to this category analyze software systems and, in addition to providing feedback on the specific errors found in terms of usability, are given the option of correcting them automatically. Tools that perform error correction in a fully automated manner (without prior validation by the tool's user) or semi-automatically (the user authorizes the corresponding automatic correction) are considered in this category. In this category, there is only the Kobold tool [PS5].

5 Validity Threats

The first threat to validity is bias in the article selection process. The articles found with the search string used were evaluated according to the defined inclusion and exclusion criteria. Other researchers may have evaluated the publications differently. To corroborate the concordance in the selection of studies, meetings were held between the researchers to check the discarded preselected articles. Another aspect related to the selection of primary studies is the declared scope of our research since we only consider works published between 2016 and 2021. We may have missed some articles directly related to our research by only considering this period. We only consider the Scopus, IEEE Xplore, and WoS databases for the SMS performed. Although we found many results, more tools could have been reported in other databases. Another point regarding the scope of our research is that we do not consider the grey literature, which will most likely include results that are in line with the objective of this work.

6 Conclusions

A conclusion will be delivered according to each research question.

RQ1: What are the automated tools that support the usability evaluation?

According to the SMS carried out, it was possible to know the general panorama of the tools that support usability evaluation automatically reported in the literature. Between 2016 and 2021, 15 studies were found, of which 14 tools were identified. The reported tools show different approaches to support the usability evaluation. Note that these are presented with different methodologies and ways to support the evaluation of automated usability. The variety of reported tools spans desktop, mobile, and web applications (focused on desktop and mobile) that can be evaluated.

RQ2: Which usability evaluation-related techniques benefit from automated tools?

The most used technique is interaction recording, which makes sense since one of the most used approaches in tools is to perform usability tests to record the interaction of users with the evaluated interfaces. It can be noted that the tools focus on the methodology

they use according to the usability evaluation techniques. Some techniques reported [13, 14] were widely used (e.g., standards conformance inspection, consistency inspection), as well as techniques that were not addressed (e.g., collaborative usability inspection [28], pluralistic walkthrough [29]). This highlights that there is still work to be done to develop tools that support automatic usability evaluation. Covering the techniques that have not yet been addressed is a reason to encourage their development.

RQ3: What are the existing problems and challenges of using automated tools for usability evaluation?

According to what was identified in the primary studies, specific challenges, limitations, and problems can be highlighted when implementing the tools. One of these challenges is user interface event detection. This makes sense because it is the most important part of usability testing. Problems in detecting events in usability tests can cause erroneous results, which translates into poor usability for the evaluated interface. A similar aspect is that of the detection of indicators and thresholds. These must be defined and validated so that the tools, which focus on these aspects, can deliver a correct evaluation of usability. Although the tools greatly help the work of implementing an automated usability evaluation, usability experts are still needed, in some cases, to review the results [PS3, PS15].

RQ4: How can automated tools for usability evaluation be classified?

According to the analysis carried out on the primary studies identified in the SMS, the tools can be classified according to their approach and the functionalities that support automated usability evaluation. The classification of the tools is: (i) measure usability, (ii) support usability evaluation, (iii) detect usability problems, and (iv) correct usability problems.

The classification that includes the most tools correspond to those that detect usability problems. This is because it is a broader scope when facing a usability evaluation. These tools provide recommendations that guide the developers of the evaluated applications to correct the usability errors detected. A broader scope is that the tool automatically detects usability problems; only one tool belongs to this category. Kobold [PS5] is presented as one of the most exciting tools because it integrates automatic and semi-automatic refactoring of web user interface elements.

As future works, we will consider more databases (e.g., ACM Digital Library, SpringerLink). Additionally, study and consider the grey literature to expand the results when looking for tools that support the usability evaluation automatically. We want to explore tools that perform qualitative usability evaluations [30]. Finally, we expect to study the usability evaluations results to prioritize and recommend the most relevant aspects [31].

Acknowledgment. This work was supported by the Chilean Ministry of Education and the University of Atacama (ATA1899 project).

Appendix A: Primary Studies

This appendix lists the references of the primary studies used for the mapping study described in this paper.

[PS1] Gonçalves, L. F., Vasconcelos, L. G., Munson, E. V., Baldochi, L. A.: Supporting adaptation of web applications to the mobile environment with automated usability evaluation. In: 31st Annual ACM Symposium on Applied Computing (SAC'16), ACM, Pisa, Italy, pp. 787–794 (2016). https://doi.org/10.1145/2851613.2851863.

[PS2] Assila, A., de Oliveira, K. M., Ezzedine, H.: An environment for integrating subjective and objective usability findings based on measures. In: 2016 IEEE Tenth International Conference on Research Challenges in Information Science (RCIS'16), IEEE, Grenoble, France, pp. 1–12 (2016). https://doi.org/10.1109/RCIS.2016.7549320.

[PS3] Grigera, J., Garrido, A., Rivero, J. M., Rossi, G.: Automatic detection of usability smells in web applications. International Journal of Human-Computer Studies 97, 129–148 (2017a). https://doi.org/10.1016/j.ijhcs.2016.09.009.

[PS4] Paternò, F., Schiavone, A. G., Conti, A.: Customizable automatic detection of bad usability smells in mobile accessed web applications. In: 19th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI'17), ACM, Vienna, Austria, article 42, pp. 1–11 (2017). https://doi.org/10.1145/3098279.3098558.

[PS5] Grigera, J., Garrido, A., Rossi, G.: Kobold: web usability as a service. In: 2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE'17), Urbana, IL, USA, pp. 990–995 (2017). https://doi.org/10.1109/ASE.2017.8115717.

[PS6] Soui, M., Chouchane, M., Gasmi, I., Mkaouer, M. W.: PLAIN: PLugin for predicting the usability of mobile user interface. In: 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP'17) - Vol. 1: GRAPP, Porto, Portugal, pp. 127–136 (2017). https://doi.org/10.5220/0006171201270136.

[PS7] Desolda, G., Gaudino, G., Lanzilotti, R., Federici, S., Cocco, A.: UTAssistant: A web platform supporting usability testing in italian public administrations. In: 12th Biannual Conference of the Italian SIGCHI Chapter (CHItaly'17), Cagliari, Italy, pp. 138–142 (2017).

[PS8] Federici, S., Mele, M. L., Lanzilotti, R., Desolda, G., Bracalenti, M., Meloni, F., Gaudino, G., Cocco, A., Amendola, M.: UX evaluation design of UTAssistant: A new usability testing support tool for italian public administrations. In: Kurosu M. (ed.) Human-Computer Interaction. Theories, Methods, and Human Issues. HCI 2018 (55–67). Lecture Notes in Computer Science, vol 10901. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91238-7_5.

[PS9] Federici, S., Mele, M. L., Bracalenti, M., Buttafuoco, A., Lanzilotti, R., Desolda, G.: Bio-behavioral and self-report user experience evaluation of a usability assessment platform (UTAssistant). In: 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP'19) - Vol. 2: HUCAPP, Prague, CZ, pp. 19–27 (2019).

[PS10] Marenkov, J., Robal, T., Kalja, A.: Guideliner: A tool to improve web UI development for better usability. In: 8th International Confe- rence on Web Intelligence, Mining and Semantics (WIMS'18), ACM, Novi Sad, Serbia, article 17, pp. 1–9 (2018). https://doi.org/10.1145/3227609.3227667.

[PS11] Chettaoui, N. Bouhlel, M. S.: I2Evaluator: An aesthetic metric-tool for evaluating the usability of adaptive user interfaces. In: Hassanien, A. E., Shaalan, K., Gaber, T., and Tolba, M. F. (eds.) Proceedings of the International Conference on Advanced Intelligent Systems and Informatics. AISI 2017 (374–383). Advances in Intelligent Systems and Computing, vol 639. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-64861-3_35.

[PS12] Barra, S., Francese, R., Risi, M.: Automating mockup-based usability testing on the mobile device. In: Miani, R., Camargos, L., Zarpelão, B., Rosas, E., and Pasquini, R. (eds.) Green, Pervasive, and Cloud Computing. GPC 2019 (128–143). Lecture Notes in Computer Science, vol 11484. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-19223-5_10.

[PS13] Liu, Z., Chen, C., Wang, J., Huang, Y., Hu, J., Wang, Q.: Owl eyes: Spotting UI display issues via visual understanding. In: 35th IEEE/ACM International Conference on Automated Software Engineering (ASE'20), ACM, Virtual Event Australia, pp. 398–409 (2020). https://doi.org/10.1145/3324884.3416547.

[PS14] Bacíková, M., Porubän, J., Sulír, M., Chodarev, S., Steingartner, W., Madeja, M.: Domain usability evaluation. Electronics 10(16), 1–28, article 1963, (2021). https://doi.org/10.3390/electronics10161963.

[PS15] Al-Sakran, H. O. Alsudairi, M. A.: Usability and accessibility assessment of saudi arabia mobile E-government websites. IEEE Access 9, 48254–48275 (2021). https://doi.org/10.1109/ACCESS.2021.3068917.

References

1. Nielsen, J.: Usability engineering. Morgan Kaufmann Publishers Inc., San Francisco (1994).ISBN: 978-0080520292
2. ISO 9241–11:2018. Ergonomics of human-system interaction–part 11: Usability: Definitions and concepts, ISO (2018)
3. Ferré, X.: Marco de integración de la usabilidad en el proceso de desarrollo software. Facultad de Informática, Universidad Politécnica de Madrid, Madrid, Spain, Tesis doctoral (2005)
4. Ivory, M.Y., Hearst, M.A.: The state of the art in automating usability evaluation of user interfaces. ACM Comput. Surv. **33**(4), 470–516 (2001). https://doi.org/10.1145/503112.503114
5. Marenkov, J., Robal, T., Kalja, A.: Guideliner: a tool to improve web UI development for better usability. In: 8th International Conference on Web Intelligence, Mining and Semantics (WIMS 2018), ACM, Novi Sad, Serbia, article 17, pp. 1–9 (2018). https://doi.org/10.1145/3227609.3227667
6. Fabo, P., Durikovic, R.: Automated usability measurement of arbitrary desktop application with eyetracking. In: 2012 16th International Conference on Information Visualisation, IEEE, Montpellier, France, pp. 625–629 (2012). https://doi.org/10.1109/IV.2012.105
7. Federici, S., et al.: UX evaluation design of UTAssistant: a new usability testing support tool for Italian public administrations. In: Kurosu, M. (ed.) HCI 2018. LNCS, vol. 10901, pp. 55–67. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91238-7_5
8. Grigera, J., Garrido, A., Rossi, G.: Kobold: web usability as a service. In: 2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE 2017), IEEE, Urbana, IL, USA, pp. 990–995 (2017). https://doi.org/10.1109/ASE.2017.8115717

9. Liyanage, N. L., Vidanage, K.: Site-ability: a website usability measurement tool. In: 2016 Sixteenth International Conference on Advances in ICT for Emerging Regions (ICTer'16), IEEE, Negombo, Sri Lanka, pp. 257–265 (2016). Doi: https://doi.org/10.1109/ICTER.2016.7829929
10. Charfi, S., Trabelsi, A., Ezzedine, H., Kolski, C.: Widgets dedicated to user interface evaluation. Int. J. Hum.-Comput. Interact. **30**(5), 408–421 (2014). https://doi.org/10.1080/10447318.2013.873280
11. Bakaev, M., Mamysheva, T., Gaedke, M.: Current trends in automating usability evaluation of websites: can you manage what you can't measure? In: 2016 11th International Forum on Strategic Technology (IFOST 2016), Novosibirsk, Russia, pp. 510–514. IEEE (2016). https://doi.org/10.1109/IFOST.2016.7884307
12. Khasnis, S. S., Raghuram, S., Aditi, A., Samrakshini, R. S., Namratha, M.: Analysis of automation in the field of usability evaluation. In: 2019 1st International Conference on Advanced Technologies in Intelligent Control, Environment, Computing & Communication Engineering (ICATIECE 2019), Bangalore, India, pp. 85–91. IEEE (2019). https://doi.org/10.1109/ICATIECE45860.2019.9063859
13. Ferré, X., Juristo, N., Moreno, A.M.: Deliverable D.5.1. selection of the software process and the usability techniques for consideration. STATUS Project (code IST-2001–32298) financed by the European Commission from December of 2001 to December of 2004 (2002). http://is.ls.fi.upm.es/status/results/STATUSD5.1v1.0.pdf
14. Ferré, X., Juristo, N., Moreno, A. M.: Deliverable D.5.2. specification of the software process with integrated usability techniques. STATUS Project (code IST-2001–32298) financed by the European Commission from December of 2001 to December of 2004 (2002). http://is.ls.fi.upm.es/status/results/STATUSD5.2v1.0.pdf
15. Kitchenham, B.A., Budgen, D., Brereton, O.P.: Using mapping studies as the basis for further research–a participant-observer case study. Inf. Softw. Technol. **53**(6), 638–651 (2011). https://doi.org/10.1016/j.infsof.2010.12.011
16. Zhang, H., Babar, M.A., Tell, P.: Identifying relevant studies in software engineering. Inf. Softw. Technol. **53**(6), 625–637 (2011). https://doi.org/10.1016/j.infsof.2010.12.010
17. Assila, A., de Oliveira, K. M., Ezzedine, H.: An environment for integrating subjective and objective usability findings based on measures. In: 2016 IEEE Tenth International Conference on Research Challenges in Information Science (RCIS 2016), Grenoble, France, pp. 1–12. IEEE (2016). https://doi.org/10.1109/RCIS.2016.7549320
18. Barra, S., Francese, R., Risi, M.: Automating Mockup-based usability testing on the mobile device. In: Miani, R., Camargos, L., Zarpelão, B., Rosas, E., Pasquini, R. (eds.) GPC 2019. LNCS, vol. 11484, pp. 128–143. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-19223-5_10
19. Paternò, F., Schiavone, A. G., Conti, A.: Customizable automatic detection of bad usability smells in mobile accessed web applications. In: 19th International Conference on Human-Computer Interaction with Mobile Devices and Services (Mo-bileHCI 2017), Vienna, Austria, article 42, pp. 1–11. ACM (2017). https://doi.org/10.1145/3098279.3098558
20. Atlas.ti9 Atlas.ti 9 desktop trial (windows) (2021). https://atlasti.com/
21. Scopus.com: An eye on global research: 5000 Publishers. Over 71 M records and 23,700 titles 2020. https://www.scopus.com/freelookup/form/author.uri. Accessed 16 Sept 21
22. Castro, J. W., Acuña, S. T.: Comparativa de selección de estudios primarios en una revisión sistemática. In: XVI Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2011), A Coruña, España, pp. 319–332 (2011). http://hdl.handle.net/10486/665299. Accessed 16 Sept 21
23. Magües, D., Castro, J.W., Acuña, S.T.: Usability in agile development: a systematic mapping study. In: XLII Conferencia Latinoamericana de Informática (CLEI 2016), Valparaíso, Chile, pp. 677–684. IEEE (2016). https://doi.org/10.1109/CLEI.2016.7833347

24. Ren, R., Castro, J.W., Acuña, S.T., De Lara, J.: Evaluation techniques for chatbot usability: a systematic mapping study. Int. J. Software Eng. Knowl. Eng. **29**(11n12), 1673–1702 (2019). https://doi.org/10.1142/S0218194019400163
25. Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., Carey, T.: Human-Computer Interaction. Concepts and Design. Addison-Wesley, Harlow (1994). ISBN: 978–0201627695
26. Shneiderman, B.: Designing the User Interface: Strategies for Effective Human-Computer. Pearson, Boston (1998). ISBN: 978–0201694970
27. Hix, D., Hartson, H.R.: Developing User Interfaces: Ensuring Usability Through Product & Process. Wiley, New York (1993).ISBN: 978-0471578130
28. Constantine, L.L., Lockwood, L.A.: Software for use: A Practical Guide to the Models and Methods of Usage-Centered Design. Addison-Wesley Professional, New York (1999).ISBN: 978-0321773722
29. Nielsen, J.: Usability inspection methods. In: Conference Companion on Human Factors in Computing Systems (CHI 1994), Boston, Massachusetts, USA, pp. 413–414. ACM (1994). https://doi.org/10.1145/259963.260531
30. Rojas P., L.A., Truyol, M.E., Calderon Maureira, J.F., Orellana Quiñones, M., Puente, A.: Qualitative evaluation of the usability of a web-based survey tool to assess reading comprehension and metacognitive strategies of university students. In: Meiselwitz, G. (ed.) HCII 2020. LNCS, vol. 12194, pp. 110–129. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49570-1_9
31. Rojas, L.A., Macías, J.A.: Toward collisions produced in requirements rankings: a qualitative approach and experimental study. J. Syst. Softw. **158**, 110417 (2019). Article 42. https://doi.org/10.1016/j.jss.2019.110417