The Transparency of Automatic Web Accessibility Evaluation Tools: Design Criteria, State of the Art, and User Perception

The Transparency of Automatic Web Accessibility Evaluation Tools

Marco Manca

CNR-ISTI, HIIS Laboratory, marco.manca@isti.cnr.it

Vanessa Palumbo

CNR-ISTI, HIIS Laboratory, vanessa.palumbo@isti.cnr.it

Fabio Paternò

CNR-ISTI, HIIS Laboratory, fabio.paterno@isti.cnr.it

Carmen Santoro

CNR-ISTI, HIIS Laboratory, carmen.santoro@isti.cnr.it

Several Web accessibility evaluation tools have been put forward to reduce the burden of identifying accessibility barriers for users, especially those with disabilities. One common issue in using accessibility evaluation tools in practice is that the results provided by different tools are sometimes unclear, and often diverging. Such limitations may confuse the users who may not understand the reasons behind them, and thus hamper the possible adoption of such tools. Hence, there is a need for tools that shed light on their actual functioning, and the success criteria and techniques supported. For this purpose, we must identify what criteria should be adopted in order for such tools to be transparent and to help users better interpret their results. In this paper, we discuss such issues, provide design criteria for obtaining user-centred and transparent accessibility evaluation tools, and analyse how they have been addressed by a representative set of open, license-free, accessibility tools. We also report on the results of a survey with 138 users of such tools, aimed at capturing the perceived usefulness of previously identified transparency requirements. Finally, we performed a user study with 18 users working in the Web design or accessibility fields with the goal of receiving more feedback about the transparency of a selected subset of accessibility tools.

CCS CONCEPTS • Human-centered computing → Accessibility systems and tools.

Additional Keywords and Phrases: accessibility, automatic validation tools, transparency

ACM Reference Format:

First Author’s Name, Initials, and Last Name, Second Author’s Name, Initials, and Last Name, and Third Author’s Name, Initials, and Last Name. 2018. The Title of the Paper: ACM Conference Proceedings Manuscript Submission Template: This is the subtitle of the paper, this document both explains and embodies the submission format for authors using Word. In Woodstock ’18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY. ACM, New York, NY, USA, 10 pages. NOTE: This block will be automatically generated when manuscripts are processed after acceptance.

1 INTRODUCTION

Following the adoption of accessibility laws, many public organizations have started paying more attention to accessibility guidelines. In several countries there are various efforts to promote best accessibility practices (see for example [Gulliksen et al., 2010], [Lazar and Olalere, 2011]), and initiatives such as the European Accessibility Directive (WAD, 2016/2102 on the accessibility of the websites and mobile applications of public sector bodies) [EU Commission, 2016], have further promoted the right of disabled people to have access to online public information and services. However, Web accessibility requires constant monitoring of numerous details across many pages of a given site. Thus, to simplify the monitoring, analysis, detection, and correction of website accessibility problems, several automatic and semi-automatic tools have been proposed. Even though accessibility validation is a process that cannot be fully automated [Vigo et al., 2013; Power et al., 2012], automatic tools still play a crucial role in assessing the accessibility of websites, some also using relevant metrics [Brajnik and Vigo, 2019]. Such tools help human operators collect and analyse data about the actual application of accessibility guidelines, detect non-compliance, and provide information about how to address the possible problems. Over time many tools have been put forward in this area (e.g. [Beirekdar et al. 2002], [Beirekdar et al. 2005], [Gay and Qi Li 2010], [Fernandes et al. 2014], [Kasday 2000], [Schiavone and Paternò 2015]). Another significant effort in this regard was the European Internet Inclusion Initiative Checker (EIII Checker), which was designed and implemented as part of the EIII project supported by the European Commission [Mucha et al. 2016], [Nietzio et al. 2011]). As of October 2021, the W3C Web Accessibility Evaluation Tools list [W3C WAETL] contains 159 software packages. However, for various reasons most accessibility tools had limited impact, and often they are not able to keep up with the latest technology and become rapidly obsolete [Paternò and Schiavone, 2015]. Some are just local efforts that do not even support the English language (such as Hera FFX [Fuertes et al. 2009] and Vamolà [Mirri et al. 2011]). Other tools only focus on limited accessibility aspects. For example, [Miniukovich et al. 2019] have focused on readability issues, and provide some automatic support but only for assessing some text-related properties, while the A11y Color Contrast checker [A11y] mainly focuses on checking the contrast in a Web page. Another example is one proposal [Moreno et al. 2019], which does not aim at validating accessibility guidelines, but it focuses on providing some automatic support for simplifying textual expressions. As [Abascal et al. 2019] indicate, in general, accessibility validation tools can be classified according to various criteria: the type of license (free versus commercial); the platform where they can be executed; the evaluation scope (ranging from single pages to entire websites); the support provided for repairing identified issues; how the evaluation results, guidelines supported and detected issues are rendered, and also exported.

While observing and discussing with users of such tools and developers of other tools, we often noticed that they differ in their coverage of accessibility guidelines, in how they interpret and to what extent they are able to support them, and in the design of how they present the results, including errors (and likely errors that may need human intervention to be actually evaluated). In order to facilitate the management of the accessibility guidelines validation, [Arrue et al. 2008] put forward an XML-based language for guidelines definition, which uses XPath sentences to implement the guidelines. In this perspective, [Pelzetter 2021] proposes a declarative model, represented by the Web Ontology Language (OWL), for describing the requirements for accessible web pages in terms of ACT Rules [ACT Rules 2021], which are aimed at helping developers of accessibility evaluation tools to create tools that produce more consistent results and are easier to maintain, but still have limited coverage of the possible accessibility issues. Still in this regard, [Brajnik et al., 2012] also indicate that evaluating the conformance to accessibility guidelines is a process on which it is difficult to achieve easy agreement between evaluators. The aforementioned differences are also the reason for different results obtained by different validators on the same Web content. Moreover, these differences result in being perceived in different ways by users; sometimes they are misinterpreted, and can generate misunderstandings and lack of trust in automatic validation tools.

Clearly exposing how a tool supports such features can better assist end users, website commissioners, designers and developers in making informed decisions, and indicate gaps that could be addressed in future versions of these tools (or in new tools). Unfortunately, this issue has not been sufficiently dealt with in previous studies of accessibility tools. We thus introduced [Parvin et al. 2021] the concept of transparency of such tools, as well as some criteria that can be used to analyse it, and provided an initial analysis of four validation tools according to them. In general, by transparency of an accessibility validation tool, we mean its ability to clearly indicate to its users its actual inputs for the validation (the guidelines and techniques supported), the meaning of the results generated and its limitations. The contribution of this paper is to present a refined definition of the design criteria that can be used to assess the transparency of accessibility validation tools, analyse how a set of open, license-free tools support them, and report on user feedback obtained in a survey and a user test conducted to validate the relevance of the criteria identified, and use them to drive the transparency assessment of some validation tools.

The various available tools follow different approaches to checking accessibility. They have to keep up with the continuous evolution of Web technologies and their use, which imply an evolution of the accessibility guidelines, and in turn the need to continuously update their support for the validation. Moreover, the W3C WCAG accessibility guidelines are defined in a format that must be interpreted by tool~~s~~ developers to actually implement them. Users are sometimes not even aware of such aspects, and they may become disoriented when they see different tools providing different results in terms of validation. Thus, it is important to make them more aware of such issues and provide tool developers with indications for making their accessibility tools more transparent.

To some extent, we face similar issues to those that people are encountering with the increasing deployment of Artificial Intelligence (AI) tools, which often generate problems for their users since they do not explain why they are operating in a certain way. Thus, interest in techniques for explainable AI has been increasing recently. In this perspective, some researchers have explored the space of user needs for explanations using a question-driven framework. For example, some authors [Liao et al. 2020] propose a question bank in which user needs for explainability are represented as prototypical questions, which users might ask about the AI, such as “Why is this instance given this prediction?”, “What would the system predict if this instance changes to …?” Some of such questions can still be relevant for accessibility validation tools, even when they do not use AI methods (of course, for accessibility validation tools the meant “predictions” will concern the verification of accessibility criteria).

In this paper, after an analysis of relevant work in the state of the art (Section 2), we redefine the transparency concept and indicate some design criteria to support it (Section 3), and provide an analysis of a set of accessibility validation tools according to such criteria (Section 4). Then, we investigate users’ perception of tools transparency through a survey (Section 5) and a user test (Section 6), and finally conclude with some discussion and recommendations for future work.

To analyse how a set of validation tools support the transparency criteria, we selected eleven tools from the W3C list available at [W3C WAETL], which enable testing Web pages against the WCAG 2.1 guidelines, and are non-commercial (not licensed) and freely accessible on the Web. There are commercial and licensed tool that provide a good level of accessibility evaluation, however they are not included in our study because they are unavailable to the general public, while we believe accessibility tools should be available to all.

The empirical feedback on the transparency issues has been obtained through a survey and a user test. The survey allowed us to analyse the aspects and functionalities that an accessibility evaluation tool should support in order to be transparent towards its users, while the test provided an opportunity to assess how users directly experience the current transparency status of a set of representative accessibility validation tools. The survey and the user test have helped us answer the research questions concerning whether the transparency criteria identified in this paper are representative for users involved in the accessibility field/process, and to what extent current accessibility validation tools are perceived as transparent, according to such criteria.

The survey collects the opinion of 138 people who have used such tools and are, to some extent, involved in accessibility. They are classified according to three roles: Web commissioners (people who mainly decide and manage the content of a Web site), accessibility experts (those who are in charge of actually checking whether an application is accessible), and Web developers (those who actually apply the accessibility best practices in creating Web sites). The user test involved eighteen people who were asked to perform a typical set of tasks in accessibility validation using three different tools, and then had to rate transparency aspects of each tool. The final discussion highlights important aspects to consider based on the empirical data, the users' suggestions, and the analysis carried out.

2 RELATED WORK

Interest in automatically supporting accessibility validation started several years ago, and various contributions have analysed existing validation tools from different perspectives. For example, [Brajnik 2004] has discussed the effectiveness of accessibility evaluation tools in terms of completeness, correctness, and specificity. [Ivory et al. 2003] put forward an initial exploratory analysis of automated evaluation and transformation tools to help Web developers build better sites for users with diverse needs, and found that there are large categories of users whose needs are not yet adequately addressed. [Molinero et al. 2006] conducted a study showing that the results provided by Web accessibility tools are often variable, and thus users may conclude that they are not reliable. [Petrie et al. 2007] reported on a usability evaluation of five accessibility evaluation tools. A group heuristic evaluation was conducted, with five experts in usability and accessibility working through each tool together, but rating usability problems separately. The results showed that the usability of these tools was limited and that they do not support Web developers adequately in checking the accessibility of their Web resources. In a more recent survey [Yesilada et al. 2015], respondents strongly agreed that accessibility must be grounded on user-centred practices and that accessibility evaluation is more than just inspecting source code. Generally, it is easy to see that when applying different validation tools to the same Web content, they provide different results, and users have difficulties understanding the reasons for such variability, and to what extent the results are meaningful. Thus, also for improving their usability, there is a need for more transparency to help users better interpret their results [Parvin et al. 2021]. In another work, [Vigo et al. 2013] analysed the effectiveness of six frequently used accessibility evaluation tools in terms of coverage, completeness, and correctness with respect to the WCAG 2.0 guidelines. They found that coverage was narrow as, at most, 50% of the success criteria were covered, and similarly, completeness ranged between 14% and 38%; however, some of the tools that exhibit higher completeness scores produce lower correctness scores (66-71%) because catching as many violations as possible can lead to an increase in false positives. Lastly, they indicated that the effectiveness in terms of coverage and completeness could be boosted if the right combination of tools is employed for each success criteria. A further study on automatic Web accessibility evaluation [Abduganiev 2017], which only considered support for the WCAG 2.0 guidelines, has analysed eight popular and free online automated Web accessibility evaluation tools finding significant differences in terms of various aspects (coverage, completeness, correctness, validity, efficiency and capacity). A study [Ballantyne et al. 2018] has considered a set of guidelines for mobile app accessibility, and applied them to a set of Android apps, but the validation was carried out in a completely manual manner through a kind of heuristic evaluation exercise, which can provide limited information and requires particular effort. More recently, [Padure et al.

2019] compared five automatic tools for assessing accessibility. The result of the study indicates that the combined use of two of the considered tools would increase the completeness and reliability of the assessment. [Frazao and Duarte 2020] focused their analysis of accessibility on validation plugins extensions for the Chrome Web browser. They found that individual tools still provide limited coverage of the success criteria, and the coverage of success criteria varies quite a lot among different evaluation engines. After analysing their results, they recommend using more than one tool and complementing automated evaluation with manual checking. [Burkard et al. 2021] compared four commercial monitoring accessibility tools. In this study, the tools were evaluated based on several criteria such as coverage of the Web pages, success criteria, completeness, correctness, support for localisation of errors, and manual checks. However, none of such studies focused on transparency aspects and how to help users understand how the accessibility evaluation tools work.

In general, little attention has been paid to how the automatic accessibility evaluation tools should be designed to provide clear information about their coverage and working, and we aim to provide indications about how to address such aspects. As we mention in the introduction, there are some similarities between transparency in accessibility validation tools and in artificial intelligence systems. [Liao et al. 2020] have provided a XAI Question Bank with leading questions for supporting their explainability. The first aspects that they indicate as relevant are the input and the output of the system in order to provide answers to questions regarding: What kind of data does the system learn from? and What kind of output does the system give? The criteria that we propose aim to address these aspects as well as questions such as Why/How is this instance given this prediction? and How should this instance change to get a different prediction? [Hellmann et al. 2022] have used such questions in the development of an instrument for measuring users' perception of transparency in recommender systems. Introducing support for these concepts in the accessibility validation tools will provide users with more awareness about their capabilities, and this will enable the opportunity to obtain more accessible web sites.

3 ASPECTS RELEVANT FOR TRANSPARENCY OF ACCESSIBILITY VALIDATION TOOLS

In our view, an accessibility tool is defined as *transparent* when it clearly provides its end users with proper information about what the tool supports (i.e. set of guidelines, success criteria, techniques) and is actually able to check, and also what it is not able to check, the results it produces (at different levels of granularity, i.e. from checking the basic elements of a Web page to overall measures of the accessibility of the considered pages), and how they are obtained, as well as further indications about the steps that users should take to solve the identified accessibility issues, also considering that they may be seen by people with different roles and expertise. As such, being "transparent" for a tool does not mean providing users with all the details about its inner algorithms and data structures; rather, it means providing the various stakeholders with comprehensible information that is relevant for them to understand the tool's actual capabilities, and use them effectively. The transparency criteria we propose have been derived from a previous preliminary analysis [Parvin et al. 2021] and direct experience with such tools in research and development projects, collaboration with the national agency for accessibility, teaching accessibility validation in HCI courses and, more generally, with the analysis of current accessibility validation practices, and observations of and feedback gathered from interaction with accessibility experts.

For such reasons, in order to implement the transparency definition described above, an automated validation tool should make explicit the following information on its operations, concretized in the following design criteria:

- *C1: What standards, success criteria, and techniques are supported*. This point is critical because it helps clarify the reasons for the different results in different tools. Moreover, the more techniques a tool actually covers, the more complete the results are, because different techniques reveal different accessibility problems. Indeed, the WCAG 2.1 guidelines are composed of 78 success criteria and many associated techniques (which have increased over the years), and some of them cannot be automatically validated at all. Thus, users need to understand the current coverage of the considered tools, especially in terms of the specific techniques supported, since they drive the actual results.

- *C2: How accessibility issues are categorized*. The classification of the accessibility validation results indicated by the EARL W3C standard [Abou-Zahra 2017] recommends using one of the following categories to indicate the tests' results: passed, failed, cannot tell, inapplicable, and untested. The more a tool utilises this standard classification for the accessibility issues, the more understandable its results will be for users. If a tool uses different terms to categorize its accessibility results, their explanation must be clear and readily available to users, along with how they refer to the standard categorization.
- *C3: How the validation results are provided by the tool.* For this purpose, it is important that such information is provided with varying granularity levels, and using different types of presentations.
    - *C3a: Granularity*. The tool should be capable of providing indications for accessibility of specific elements but also overall accessibility measures, for entire Web pages or sites. In addition to reporting lists of detailed issues, the use of overall metrics can help indicate the overall accessibility level of the considered websites. These metrics can be defined based on success criteria, and the corresponding sufficient, advisory and failure techniques. Some non-technical users may be interested in high abstraction level results; thus, metrics and tables can help such users get a general understanding of the accessibility results.
    - *C3b: Presentation type*. There should be different ways to report validation results, so as to fulfil the needs of different types of users with diverse expertise and skills. For example, an annotated code view can be more suitable for developers, while a report with charts and statistics summarising the detected issues can be more intuitive for non-technical users, such as Web commissioners.
- *C4: Whether the tool provides practical indications about how to solve the identified problems.* Some tools are only able to evaluate web pages and do not include functionality to help users correct the identified accessibility violations. Clearly, useful additions would be to provide 'repairing' functionalities that assist users through the process of correcting some accessibility problems, or by providing suitable recommendations for solutions.
- *C5: Whether the tool is able to provide information about its limitations.* This point is critical to allow users to interpret the results correctly. One of the most representative examples in this regard is whether the tool is able to evaluate dynamic pages or not. Several accessibility validation tools still rely only on static HTML. However, current Web sites have largely evolved into more dynamic applications (with Ajax scripts, or developed with frameworks such as Angular). In this case, the absence of errors should not indicate that the target application is fully accessible, but rather that the tool is unable to access the actual version with which the user interacts. Thus, not only is the tool unable to fully assess it, but it also does not provide any indication of this limitation, with the potential risk of generating a false sense of confidence among users. In conclusion, to be transparent, the accessibility validation tools should provide their users with clear information about their full functionality, including possible limitations.

4 ACCESSIBILITY TOOL ANALYSIS AND COMPARISON

4.1 Tool Selection

In order to perform the analysis of the transparency of accessibility tools, we selected a subset of them from the W3C website section where the Web Accessibility Initiative group provides a list of evaluation tools that can be filtered according to various criteria [W3C WAETL 2021] (this list contains 159 tools as of 20 October 2021). It is possible to filter tools by guidelines, languages, type of tool (API, browser plugin, command line, online tool, etc.), depending on the possibility to evaluate single, multiple and private pages, and on the license type (commercial vs free). In order to obtain a representative set of tools we applied the following filters to the list:

- **Guidelines**: WCAG 2.1. W3C released such version in 2018, thus current evaluation tools must support them.
- **Supported language**: English. Accessibility is not a national concern, it involves the whole world and has no borders; so, evaluation tools should be accessible by the majority of the interested users by supporting at least the English language.
- **Type of tool**: Online Tools or Browser Extensions. Our goal is to evaluate websites; so, we should use online tools or extensions installed on browsers.
- **Supported Formats**: HTML and CSS. As explained before, we would like to evaluate websites; thus, validation tools should be able to evaluate at least HTML and CSS code.
- **Assist by**: Generating reports of the evaluation result. Tools should support users by providing the most general assistance support: reports of the evaluation results. Displaying information within the pages or modifying the presentation of the evaluated page are advanced features that can be useful only for a restricted segment of users, such as Web developers.
- **Automatically Checks**: Single Web pages or Groups of Web pages or websites. Tools should evaluate at least a single Web page; however, evaluating an entire website or a set of Web pages has been considered an important feature.
- **Licence**: Free software. We think that accessibility is a cornerstone upon which to build Web contents, thus its evaluation should be available to everyone without paying a licence fee.

We also decided to apply additional filtering criteria, even if not included in the ones provided by the W3C website. First, we excluded the tools which were provided only through their source code (i.e. released in software repositories such as GitHub) because such tools are only available to people with specific development skills for installing them, and thus they are not suitable for all people interested in accessibility. In the obtained list, we further discarded the tools that focus only on specific aspects (such as checking the colour contrast), and do not aim to provide general support for the WCAG guidelines, or those which ask for further information from users (e.g., email address) before actually providing the report.

By applying such filters to the list provided by the W3C website, we obtained 11 tools. The analysis has been carried out based on the publicly available versions of the considered tools in July 2021. The tools selected are:

- aCe [accessiBe], which allows users to evaluate the ADA (Americans with Disabilities Act) and WCAG compliance of a specific Web page. Users can not choose the WCAG version and the level of conformance, but the overview page reports that the tool has been designed to focus on full WCAG 2.1 AA level compliance; no information is provided about the actually supported techniques. A Web report is provided containing a general score (Compliant, Semi-Compliant and Not Compliant) that can be considered as a sort of accessibility metric.

- The Accessi.org tool [Adam], which supports validation of WCAG 2.0 and 2.1 of single pages; it does not provide any information about the supported success criteria and guidelines. It is possible to filter the validation results according to the conformance level, priority, the tag type. In some cases, the report also provides a visual example (not related to the validated page) compliant to the considered technique and another one that represents a bad example of a situation that violates the technique.
- The UserWay Accessibility Scanning & Monitoring tool [UserWay], which has two versions. The Free Scan supports only single page validation and analysis for desktop interfaces; the Pro version supports Web sites validation also for mobile interfaces. The validator supports the version 2.1; however, to understand how UserWay Accessibility Scanning & Monitoring supports the success criteria and techniques, the FAQ section suggests to contact the customer care team, whereas from a transparency viewpoint a tool should immediately expose which criteria and techniques it supports.
- EqualWeb [EqualWeb] offers several plans and services for monitoring and analysing the accessibility of websites; among them, there are two free plans, both available through the website (only after registration) and a browser plugin. There is no actual documentation explaining which standards, success criteria and techniques the tool supports. The only information is contained in one of the FAQs, which states that the tool handles all aspects of the accessibility legislation (AA level), and all subjects and guidelines of WCAG 2.1, ADA, Section 508 and EN 301549.
- The Accessibility Checker [EXPERTE] supports the multipage evaluation; the discovered Web pages are evaluated against 41 features across 8 categories (Navigation, Aria, Names and Labels, Contrast, Tables and Lists, Best Practices, Audio and Video, Internationalization and Localization). The evaluation done by this tool exploits the Google's Lighthouse open source tool.
- The Free Web Accessibility Check [AlumniOnline Web Services], which mainly provides a plugin to address accessibility problems on WordPress websites; in addition, it also offers a scan for single pages. The plugin supports the section 508 and WCAG 2.1 level A/AA standards by providing a list of 71 accessibility issues that it addresses; unfortunately, there is no relation between such detected issues and the corresponding WCAG technique
- [IBM Equal Access Accessibility Checker] is an open-source browser extension for Web developers and auditors which, by utilizing IBM's rule engine, detects accessibility issues for Web applications, helping users identify the source of accessibility issues and try fixes. The supported accessibility guidelines are among IBM Accessibility, WCAG 2.1 (A, AA) and WCAG 2.0 (A, AA).
- The [MAUVE++] accessibility evaluator is both provided as an on-line tool and as a browser plugin. It is able to validate websites against WCAG 2.0 and 2.1 for levels A, AA, AAA. Currently, it supports 107 HTML and 8 CSS techniques and addresses 46 Success Criteria (for the WCAG 2.1)
- [QualWeb] is an open-source automated Web accessibility evaluation service that incorporates contributions from different research projects and efforts. The tool can evaluate a set of WCAG 2.1 Techniques (43 WCAG 2.1 HTML and 5 WCAG 2.1 CSS Techniques) and ACT Rules (69 in total).
- [TAW] is a free automatic online tool for ~~analyzing~~analysing website accessibility. Even though it declares to support WCAG 2.1, and for this reason it appears in the list of the considered tools, we did not find its actual support to the 2.1 guidelines, but only to the 2.0 ones. The online tool supports WCAG 2.0 level A, AA, AAA.

- [WAVE] (Web Accessibility Assessment Tool) is a free tool provided by the *Web Accessibility In Mind* (WebAIM) organization. Its functionalities are available through both a website and a browser plugin (for Chrome and Firefox) to evaluate dynamic Web content. It detects the compliance issues found in WCAG 2.0 guidelines, WCAG 2.1 guidelines (23 HTML/CSS supported techniques), and many of those in Section 508.

4.2 Comparative Analysis of the Considered tools

This section introduces a comparative analysis between the previously listed accessibility evaluation tools, by considering the transparency criteria identified in Section 3. Table 1 summarize the tools' characteristics following the above-mentioned criteria.

Regarding the accessibility support, the majority of the selected tools declare the guidelines that they are able to validate, but only five (of the eleven) explicitly indicate the techniques implemented within the tool (see Table 1). From the user perspective, the possibility to know exactly which techniques are implemented would increase transparency. In this way, users can know which accessibility aspects are covered or not by the tool, and thus which features have to be manually inspected in order to guarantee the full accessibility of the considered Web site.

Regarding providing information on how to solve the problems detected, most tools provide links to W3C documentation. The EXPERTE's Accessibility Checker provides links to WebDev.com documentation. A few tools provide some additional information: IBM Equal Access Accessibility Checker provides some recommended remedies, WAVE for each issue generates some info on what it means, why it matters and what to do.

The limitations of the tools are not clearly indicated in several cases. The list of concrete techniques addressed is provided by some tools. The clearest tools in indicating their possibilities, and consequently their limitations, are EXPERTE's Accessibility Checker, MAUVE++, QualWeb. In this perspective, the dynamic support feature is an aspect that is rarely listed among the tools' features. This aspect can be quite useful in the case of Single Page Applications (SPA) or highly dynamic Web pages populated through external services calls. It can play an important role in increasing the transparency process. If we consider a website developed through the Angular or Vue.js framework, if a tool does not support the validation of dynamic applications, its results may be wrong because they refer to unpopulated HTML representing the DOM before being loaded in the user's browser. A tool able to implement the dynamic support feature can simulate the loading phase as if the page was opened in the user's browser. In this way, the validation will be more complete than validating an almost empty page. Among the ~~analyzed~~analysed tools MAUVE++ explicitly states that it is able to support this feature. Also EqualWeb and IBM Equal Access Accessibility Checker can support it since they are also distributed as a browser plugin, which can send the actual DOM loaded in the browser to the validator, while QualWeb supports this feature because it exploits [Puppeteer], a library that launches a headless version of Chromium that can provide the complete loaded DOM to the validator.

Concerning how the validation results are reported, the most adopted solution is a summary table with numbers of issues grouped by principles or success criteria, and then lists of their occurrences. In some tools it is possible to show the corresponding code excerpt. The tools that provide richer and more flexible ways to report issues are Wave and MAUVE++. WAVE reports errors through two panes, one with summary, details, structure and contrast tabs, and one with the web page annotated with icons located where the issues occur. MAUVE++ provides two different views: Web developer view (where it shows the page code with highlighted the parts that generate issues) and End-user view, which shows errors and warnings through charts and tables.

In terms of metrics summarising the results, the *Accessibility Percentage* provided by some tools is a metric that aims to summarise how accessible a Web page/site is. In addition to such metric, MAUVE++ also offers another metric called *Accessibility Completeness*, defined as the percentage of evaluated checkpoints for which the tool has been able to make a validation with definite results (i.e. either success or failure).

The last considered feature is the Result Category; almost all the tools categorize the accessibility issues in terms of *Passed*, *Failed* and *Cannotell*; this is the terminology used by the W3C EARL standard. Cannotell denotes an uncertain outcome. This happens when an automated test requires human analysis to make a definitive decision. In some tools the Failed outcome is also indicated as an Error; the Cannotell class is also called Warning; while Passed is also called Success. Some tools (see Table 1) also include a category called Not Applicable, which denotes that the test or condition does not apply to the considered Web page, still according to the EARL standard. The tools that have results classifications substantially different from EARL are Accessi.org, Equal Web, and Alumni.Online.

Tool name	**C1- What is supported**	**C2- How accessibility issues are categorized**	**C3- How the reported information is provided**	**C4 - Info on how to solve issues**	**C5 - Info on limitations**
aCe by accessiBe https://accessibe.com/	WCAG 2.1 AA No info on techniques	Success/Failed/ Neutral Score	Accessibility Score & Compliance Level Numeric score for each web content category	No link between the checkpoints and corresponding WCAG technique.	No explicit info
Accessi.org https://www.accessi.org/	WCAG 2.0, 2.1 No info on techniques	Low or Medium impact	No metrics For each success criteria link showing the elements that generated the error, a text explaining the issue	Link to the W3C technique and positive / negative examples	No explicit info
Accessibility Scanning & Monitoring by UserWay https://userway.org/scanner	WCAG 2.1 No info on techniques	Number of tests (passed, failed, not applicable); Violations (low, medium high severity)	No metrics accessibility violations grouped by WCAG 2.1conformance level indicating the number of tests	They suggest to contact the customer care team	No explicit Info
EqualWeb https://www.equalweb.com/	WCAG 2.1, ADA, Section 508 / EN 301549 No info on techniques	General errors/Contrast errors/Notices/ Warnings/ARIA attributes/ ROLE attributes	Overall percentage accessibility Score, and Assessment Statistics, calculated on all the scanned pages Expandable with technique description and relevant code	Info on how to address the relevant technique, with link to the relevant W3C page	No explicit Info. (Browser extension for dynamic content)
Accessibility Checker by EXPERTE https://www.experte.com/accessibility	41 features across 8 categories (Navigation, Aria, Names&Labels, Contrast, Tables&Lists, Best Practices, Audio&Video, Internationalization & Localization)	Passed/failed/not applicable	Accessibility Score (derived from Lighthouse) Report with list of problems and associated failing elements	Link to documentation WebDev.com	Yes, it explicitly indicates that are able to detect a subset of issues, and indicate the types of issues addressed

Free Web Accessibility check by AlumniOnline https://www.alumnionlineservices.com/scanner/	66 accessibility issues	Errors	No metrics Report with a list of errors with a problem description and the associated code, an issue summary with the number of elements for each problem	Link to the W3C website explaining the corresponding success criteria	Partially, it lists the issues addressed
IBM Equal Access Accessibility Checker https://www.ibm.com/able/toolkit/tools	WCAG 2.1 (A, AA), US 508, EN 301 549 96 requirements	Violation/ Needs review/ Recommendation	Percentage (Percentage of elements with no detected violations or items to review) It provides summary information in terms of the percentage of elements with no detected violations. By selecting each specific issue, it is possible to get further information about the associated technique(s), the concerned element in the code.	It provides some recommended remediations	Partially, it lists the addressed requirements(Browser extension for dynamic content))
MAUVE++ https://mauve.isti.cnr.it/	WCAG 2.0, 2.1, some ACT Rules, 107 HTML, 8 CSS techniques	Errors/Warning/ Success/Not Applicable	Accessibility Percentage/Accessibility Completeness it provides two different views: Web developer view (code-oriented style) and End-user view, which shows errors and warnings through charts and tables.	Link to the W3C website explaining the corresponding Technique	Yes, it explicitly indicates that are able to detect a subset of issues, and indicate the types of issues addressed
QualWeb http://qualweb.di.fc.ul.pt/	WCAG 2.1 43 HTML, 5 CSS Techniques ACT Rules (69 in total)	Passed/Failed/ Warning/Not Applicable	No metrics. The report consists of several sections: The summary shows the total number of errors. It also allows users to filter the results that match their particular needs. The tool's report includes the description and the results of the tested rules	Link to the W3C description of the rule and, the related success criteria	Yes, it explicitly indicates that are able to detect a subset of issues, and indicate the types of issues addressed
TAW https://www.tawdis.net/	WCAG 2.0 (A, AA, AAA) No info on techniques	Problems, Warnings, Not Reviewed	No metrics Summary report with number of problems, warnings, not reviewed grouped by WCAG principles	No info	No Info
WAVE https://wave.webaim.org/	WCAG 2.0, 2.1, Section 508 (U.S accessibility law) 23 HTML/CSS Techniques	Errors/Alerts/Features/Structural elements/ HTML5/ARIA/ Contrast errors	No metrics It reports errors through two panes, one with summary, details, structure and contrast tabs. One with the web page annotated with icons located where the issues occur	Yes, when selecting on an icon it generates some info on what it means, why it matters and what to do	It does not provide clear indications about limitations. It supports a browser extension for validating dynamic content

Table 1: Tool Transparency features

5 THE SURVEY

We conducted a survey to gather a better understanding of opinions and expectations of users about the topic of transparency of accessibility validation tools. In particular, the main objective of the survey was to understand how the design criteria for the transparency of such tools, which we present in Section 3, were considered relevant and useful by various types of users of such tools.

It was an online survey, distributed to a number of groups/facebook pages such as Web Design & Development group, Web Developers @webdev4u page; Web Accessible Web @accessible.Web page; Accessibility World – Web, Matters @weba11ymatters page, Accessibility Partners @accessibilitypartners page; Accessible Web @accessible.websites page; Web Accessibility @wai4pwd page, and mailing lists of relevant projects (e.g. EU H2020 Wadcher) and associations (i.e. Hcitaly, Eusset, Bcs-hci, Chi-announcements). In addition, we also directly contacted single experts working in the accessibility field. The survey remained active for more than 3 months. In the survey, we used open questions as well as ratings that participants had to provide using a 5-point scale (where 1 was the most negative score and 5 was the most positive one, i.e. 1=not useful at all; 5= very useful), and in which only the extremes were explicitly labelled. The survey was distributed as a Google Form document, and users accessed it through the corresponding link.

5.1 Structure of the Questionnaire

The survey was structured into the following parts:

- A socio-demographical section (gender, age, country of origin, the sector in which the user works, the number of employees of his/her organization, the role that the user plays in it);
- A question asked information about whether the user exploits or not automatic accessibility assessment tools for his/her work, and, if yes, how often and which ones;
- A section asked how the user would define the transparency of automatic accessibility assessment tools. In addition, it asked, on a scale from 1 (=not useful at all) to 5 (=very useful), how useful the user rates the following features (closely related to the criteria introduced in section 3) in automated accessibility validation tools, in terms of transparency:
    - S1- That the tool states what standards, success criteria and techniques it supports in the assessment (Criterion C1);
    - S2 - That the tool specifies how it categorizes evaluation results (errors, warnings, etc.), (Criterion C2);
    - S3 - That the tool is able to provide general measures that make explicit the level of accessibility of the website/mobile app (Criterion C3);
    - S4 - That the tool presents the evaluation results both in a summarized way (e.g., graphs, tables, etc.) and in a detailed way (e.g., code view) (Criterion C3);
    - S5 - That the tool gives some practical indications on how to solve the detected problem (Criterion C4);
    - S6 - That the tool gives some indication of its limitations (Criterion C5).
- Next, as a follow-up question of the last rating it was asked to provide possible examples of limitations, and then an additional question whether they have ever experienced not to be able to understand the results of an accessibility evaluation performed by an automated tool and, if yes, which kind of difficulties they found;
- Finally, a question asked about any other features the user thinks an automated accessibility evaluation tool should have to be transparent.

5.2 Participants

139 users participated in the survey. However, one user was not considered as she provided careless responses to the survey (namely: in correspondence with the open questions, that user provided meaningless sequences of characters), thereby she was excluded from the analysis. Thus, in the end, we considered 138 users. Out of them, 92 were males (66.67%), 45 were females (32.61%), 1 user identified as “other”. Table 2 provides information about the organizations where the participants work, while Table 3 provides information on the user role that more properly characterizes the participants of the survey.

Work Organization	**Percentage**	**Number**
Public administrations	38.41%	53
Private companies	25.36%	35
R&D area	24.64%	34
Freelance	5.07%	7
Education	2.17%	3
University students	1.45%	2
Others: Publishing, digital business, informatics, no-profit organization.	2.88% (in tot)	1 each

Table 2: Organizations in which participants work

As for the size of their organizations, 43 of the users’ organizations (corresponding to 31.16% of the total) have more than 1000 workers; 43 companies (31.16%) have less than 50 workers; 30 (=21.74%) have 101-500 workers; 12 organizations (8.70%) have 501-1000 workers; 10 (=7.25%) have 51-100 workers.

Role that participants have in their work	**Percentage**	**Number**
Accessibility expert	36.23%	50
Web developer	24.64%	34
Web commissioner	10.87%	15
Researcher	5.80%	8
IT manager	3.62%	5
People supporting digital transition of public administrations	2.17%	3
Student, test specialist, academic/scientist, UX/UI expert, managing digital services/systems	7.2% (tot)	2 each (10 tot)
Data protection officer; system and infrastructure administration; programmer; consultant; legal expert; Web manager, project manager; cloud engineer; inclusive designer of the environment; technician; informatics; IT worker; institutional communications worker	9.36% (tot)	1 each (13 tot)

Table 3: Role that participants have in their work

5.3 Analysis of the Answers

In this section, we ~~analyze~~analyse the answers that the users provided to the questions included in the survey, distinguishing them between those provided to the open questions, and those provided to the closed questions (S1-S6). The answers to the open questions were analysed in the following manner: they were saved in an excel file and then their content was read and searched for common themes, according to which the various questions were coded and grouped. The analysis was done by one of the authors and then reviewed by the others.

5.3.1 Answers to the Open Questions

Do you use automated accessibility assessment tools to support your work? (Y/N) If yes, which one(s)?

Ninety people answered that they use accessibility tools for their work, while the remaining 48 did not use them. Regarding the tool they use most often for their work they answered: MAUVE++ is used by 32 users (16.49%); WAVE by 27 users (13.92%); Siteimprove by 21 people (10.82%); W3C Markup Validation Service by 14 persons (7.22%), Lighthouse by 11 people (5.67%); axe by 11 users (5.67%); AChecker by 10 people (5.15%); Vamola by 10 users (5.15%), Accessibility Insights by 6 users (3.09%); IBM Equal Access Accessibility Checker by 6 users (3.09%); ARC Toolkit by 3 users (1.55%), Cynthia Says by 3 participants (1.55%), tota11y by 3 users (1.55%), WebAIM by 3 users (1.55%). Then, FAE is used by 2 users, aCe is used by 2 users, pdf checker is used by 2 users, TPGi Colour Contrast Analyzer (CCA) is used by 2 users, 2 users declared to use a mix of browsers' extensions and add-ons. Other tools (such as Jigsaw, Imergo) were mentioned by just one user. We also collected information about the frequency with which the users exploit validation tools for their job (we asked to indicate a maximum of three tools in the survey). 29 users (which correspond to the 32.22% of the 90 abovementioned users), declared to use at least one tool once a month. 26 users (28.89%) declared to use one or more tools once a day, 18 users (20%) once a year, while the remaining 17 (18.89%) once a week.

How would you define the transparency of automatic accessibility assessment tools?

The answers provided by the users are grouped according to the criteria that we identified previously. Please note that it sometimes happened that, in their responses, users mentioned more than one aspect/criterion.

What standards, success criteria, and techniques are supported. The **standard(s)** with the tool is compliant with have been mentioned by 16 users as a way to characterize the transparency of tools. Seventeen persons mentioned that when a tool explicitly highlights the **criteria** used for the evaluation and how many of them are covered, this highly contributes to increase its transparency. One user explicitly mentioned that one factor that affects the transparency of a tool is "when the tool highlights the methods and the parameters that characterize the evaluation criteria". Seven persons highlighted that one factor that affects the transparency of a tool is whether the tool highlights the **specific techniques/tests** that it applied (or not) when checking the various success criteria.

How accessibility issues are categorized. Six users mentioned this aspect, in particular the importance of having the results categorized according to different types of content (e.g. according to the implementation language, such as HTML or CSS)

How the validation results are provided by the tool. Forty-one users mentioned aspects associated with this point. In particular, the information that the tool provides to users about errors/violations of accessibility was judged an aspect that strongly characterizes the transparency of a tool. In particular, many users declared that it is important that tools provide correct and clear explanations/visualizations of such errors (possibly both in the page and in the code), and in a way that is comprehensible also by non-technical users. Furthermore, they should offer good coverage of the errors using relevant references to the page/code to better identify/localize them, and also using relevant references to the corresponding concerned criteria. Finally, they should provide clear explanations about why an issue was pointed out and what its consequences are in terms of accessibility. Other users mentioned that transparency is highly impacted by how clear the results provided by the analysis are, and also how comprehensible is the way in which the analysis is carried out, also referring to broader information on the resulting analysis, i.e. not just focusing on the errors, but indicating how the results have been obtained, the pages used for the evaluation, the clarity and comprehensibility of the results produced by the analysis and the way in which they have been obtained, the completeness of the analysis, its reliability, and verifiability/replicability.

Whether the tool provides practical indications about how to solve the identified problems. Fifteen users mentioned as a key aspect when the tools provide users with concrete **suggestions for possible solutions** to the accessibility violations identified, more precisely how and where to intervene in order to solve the identified accessibility violations/issues.

Whether the tool is able to provide information about its limitations. Another aspect that users (N=5) judged important for the transparency of the tool is that it should clearly highlight the situations that it is not able to address automatically, and therefore for which situations there is a specific **need for manual checking**. In particular, one user said that one aspect characterizing transparency is when the tool clearly states "what needs a manual validation and what would imply, in concrete terms, performing this manual validation", thus highlighting that not only it is important to remark the need for manual checking in general (as tools are never exhaustive), but also to provide guidance to the users about how to perform such manual check in concrete terms. One user highlighted that the tool is transparent when it highlights any part of the web site that the tool was not able to test. Another aspect that users (N=4) rated highly in terms of transparency was related to the **situations when the evaluations are ambiguous, or when false positive and false negatives could occur**. In such cases, one user highlighted that it would be better that the tool explained the choices made in order to arrive at the provided results. Four users highlighted that further information about the **methodology** and objectives of the tool should be provided to users, to increase transparency. Moreover, the importance of declaring the **limitations** of the assessment provided was mentioned by four users as a way to improve transparency. Among **further aspects** mentioned by participants, two of them highlighted that the inconsistencies occurring between the evaluations provided by different tools can affect transparency. One user mentioned that a tool is transparent when it is actually possible to perform some modifications to the validation results (e.g., when it is possible to declare that a "fail" is actually a "pass").

Additional aspects mentioned. Twenty-four users mentioned some more general **characteristic/quality that tools should have** in order to be transparent. Among the qualities mentioned there was ease of use, clarity, and reliability; also, the fact that the tool is free or open source, that it is independent from specific stakeholders/vendors, and that it can be used by any user in an "open" manner, and also the accountability/credibility of the organization that develops the tool. Nine users mentioned **specific features of the tool** that can affect transparency. Among them, five users highlighted the users' need to get further details and documentation about the tool, also in terms of its strengths and features. One user highlighted that the tools generally do not provide many details, thereby accessibility experts tend not to trust them. Two users mentioned that the transparency of a tool could be increased by the personalization/configuration possibilities offered by the tool.

Have you ever experienced NOT to be able to understand the results of an accessibility evaluation performed by an automated tool? 89 answered yes, 49 users answered no.

If YES, do you remember what kind of difficulties you encountered? The answers to this question are grouped according to 4 themes: results, errors, solutions, and lack of clarity.

Seventeen users mentioned difficulties connected with the **results** provided by tools. The aspects that they mentioned regarding the results can be grouped according to *5 sub-themes* (with the most frequent ones appearing first): the *mismatches* between what was reported and what the user observed, the *interpretation of results*, *divergences* between evaluations provided by other tools, *unclear/inefficient presentation* of results, *lack of completeness* in providing such results. In particular, most users complained about *mismatches*: some users reported experiencing a mismatch between what they observed and what was reported by the tool, e.g., this happened –one user said- when there was a criterion which was reported as not satisfied, whereas actually there was no error in the page. On the other hand, another user said that, although the page was not accessible, the tool said that it was. Another user highlighted that this mismatch could affect the trust that users have in tools, as sometimes users can have the feeling that the indications are not correct and therefore the tool seems buggy. Indeed, a pair of users reported having actually experienced a bug in the validation tool: "*I do recall that the technical support team were able to explain the issues and that some issues were due to bugs in the tool.*" Another point mentioned by several users regarded the *interpretation of results***:** people complained about the difficulty of understanding the results provided by the tool. One user said that sometimes the tools are limited to provide a technical summary, which makes it difficult to understand their results, especially by non-technical people. Another user said that "*it is sometimes difficult to understand what success criteria it was mapping to, why it was only picking some techniques over others, or why certain lines of code were tagged as wrong*." Another sub-theme referred to the *divergences among different evaluations*. In particular, two users highlighted that the provided results are not always in line with the results provided by other tools. A pair of users highlighted that sometimes *tools present the results in an uneasy-to-read and non-efficient manner*. In particular, one user highlighted that sometimes tools provide "*a long list of results which is not useful if you are willing to prioritize due to lack of resource*s". A final sub-theme regarded the *lack of completeness* in providing the results: one user said that she would have preferred to see in detail also the criteria that successfully passed the evaluation, and not only those which failed.

Sixteen users mentioned difficulties connected with **errors**: in many cases the reported difficulty is in understanding the reason why a point is reported as a violation of accessibility as the indications (error messages) are sometimes ambiguous, generic (i.e. do not specifically indicate where the error is), do not even actually relate to a real accessibility violation, are not always correctly associated with the concerned element, and overall are unclear and not exhaustive. One user complained about an unclear distinction between errors and warnings. Fourteen users provided further comments about reported difficulties concerning the **solutions** provided by the tools. Most of them highlighted the difficulty of having specific, clear and correct indications about how to solve the issues, reporting that currently there is a scarcity of them. A pair of users highlighted that sometimes the proposed suggestions about how to solve a specific accessibility issue are not correct or do not work. One user suggested providing "*more contextualized solutions, perhaps with small examples, to understand whether the tool has understood the context of the ~~analyzed~~analysed content or no*t". Eleven users complained, more in general, about **lack of clarity of the information provided by the tool**: sometimes tools provide messages that are generic and ambiguous, or they use a too technical language, which is not suitable for unskilled people.

Are there any other features you think an automated accessibility evaluation tool should have in order to be transparent?
42 users (30.43% of the total users) answered YES, the remaining 96 answered no.

If YES, which ones? The answers from users were grouped into 6 themes.

Seventeen users mentioned some **features and/or characteristics that the tool should have**. Aspects that were highlighted by users regarded: the possibility to evaluate also PDFs, to be open source, to include further localization possibilities (i.e. to be able to select the language to more easily understand the errors, or to be able to select a specific country, as accessibility norms can change according to them), to have the possibility to export the reports. One user suggested having a blog reporting the updates done over time on the tool, which could help –the user noted– for understanding why some evaluation results changed over time. Another user suggested making available some practical tutorials about how to write HTML and CSS. Another user highlighted that it would be important to know who is developing and releasing the tool, as well as its mission and the goals, to better evaluate its reliability and degree of confidence; the same user highlighted that another added value could be the availability of an effective support service. One user highlighted that it could be useful to know who supports/promotes the tool. One user suggested indicating how often the tool is updated. Another user highlighted that some tools (e.g., aCe) have a manual checking model that should be followed by other tools, as it helps in doing manual checks. In particular, to this regard, another user said that tools should suggest which manual tests can be done in order to verify semi-automated success criteria. A user suggested publishing the list of the tests that tools do, highlighting that some tools already do this. One user suggested proving the results delivered by the tool against a standardized set of examples, like the ones provided by ACT rules. Another user highlighted the need of solving the inconsistencies that can be found among different tools. One user declared that it would be useful to understand whether a site/app meets the WAD requirements. Another user said that tools should state outright the known statistics of how many accessibility problems can be determined through automated scans. Thirteen users mentioned the need of having **further info on the results/analysis**. Among the most relevant comments provided, one highlighted the need of having further information to precisely replicate the tests; another person suggested to clearly indicate what was not tested, and what needs to be tested manually. Another user highlighted the clarity of the results as a key aspect. A pair of users highlighted that the tool should give a **precise indication of the conformance level** (i.e. A, AA, AAA) that it considers. In addition, one of them highlighted that it would be useful **to indicate what kind of problems would be faced by people with which disability(ies)** if an error is not corrected or a criterion is not met. Another user would appreciate further information about the ARIA rules the tool evaluated. A pair of users highlighted the need to have more references to WCAG, i.e., which WCAG checkpoints are covered, how they are covered, and more code snippets with a hint on what to fix and what is missing. A user highlighted that, if the tool provides a general overall measure, it should be explicit in how it is calculated and what its limitations are. Two users emphasized the importance of having some visual indications directly in the concerned Web page (i.e., to show the layout of the Web page, to highlight key parts in it e.g., the tables). In addition, another user highlighted the importance of having further information about how the check has been done by the tool, in order to facilitate the user to verify whether there is a bug in the tool. Five users highlighted the need of providing concrete and operative **indications to solve the errors** identified, also by showing one or more examples of the solution, especially for the most common errors. Among them, one user highlighted the need of providing hands-on examples especially when specific assistive technologies are involved (i.e., screen readers), as in such situations it is not obvious that all the users of the tool know all their implications and how to solve possible problems connected with their use. Four users also pointed out the need to **provide better support to non-technical users**. This would imply, for instance, providing users with easily understandable results, possibly accompanied by visual graphs, as well as easy-to-understand explanations of the motivations why an accessibility violation was found by the tool. One user suggested having an icon that should allow users to keep track of the current state of the evaluation easily. Three users mentioned the **need to emphasize that manual check is always needed**. Users highlighted that tools should clearly state that the automatic checks they provide should in any case be complemented by manual validation. One, in particular, declared: "*They should state outright the known statistics of how many accessibility problems can be determined through automated scans and should also make clear that it is not possible for an automated tool to identify or remediate the vast majority of websites to 100% WCAG conformance.*". Finally, three users pointed out the need of **highlighting and addressing the occurrence of false positives, false negatives, and possible errors in the analysis.** They said that sometimes tools highlight issues that are not real ones or, on the contrary, could fail in identifying actual accessibility violations, thereby it would be better to solve this issue. One user, in particular, suggested that the tools should highlight the possibility of false positives in the most critical WCAG criteria. Another one suggested that **the tools should report the confidence level** associated with a specific error when it cannot be sure that it is an actual error.

5.3.2 Answers to the Closed Questions

On a scale of 1 (not very useful) to 5 (very useful), how useful do you rate the following features in automated accessibility validation tools in terms of transparency? That the tool

- states what standards, success criteria and techniques it supports in the assessment? (S1)
- specifies how it categorizes evaluation results (errors, warnings, etc.)? (S2)
- is able to provide general measures that make explicit the level of accessibility of the website/mobile app? (S3)
- presents the evaluation results in a summarized (e.g., graphs, tables) and in a detailed way (e.g., code view)? (S4)
- gives some practical indications on how to resolve the detected problem? (S5)
- gives some indication of its limitations? (S6)

	M	*SD*	*Mdn*	*IQR*	*Min*	*Max*
Standards, Success Criteria, Techniques Support	4.62	0.67	5	1	1	5
Result Categorisation	4.54	0.75	5	1	1	5
General Accessibility Measures	4.28	0.91	4.5	1	1	5
Result Presentation (Summarised vs. Detailed)	4.42	0.91	5	1	1	5
Suggestion to Solve Errors	4.67	0.74	5	1	1	5
Tool Limitations Information	4.22	0.97	5	1	1	5

Table 4: Summary table for descriptive statistics concerning users' ratings

For example, on what types of limitations the tool should provide indications (follow-up question to S6)?

The answers addressed various themes: i) the aspects that the tool is not able to automatically evaluate, ii) the preferences and parameters that users can specify for the analysis, iii) the situations in which the results can be wrong (e.g. false positive or false negatives) or ambiguous, iv) the lack of clear indications highlighting the need of manual check (with possible guidance on this manual check), and also v) further aspects.

Thirty-six users mentioned **aspects that the tool is not able to automatically evaluate** as a limitation. Many of them generically pointed out that tools should indicate the situations that they cannot automatically assess, either e.g. because they do not cover the corresponding criterion or because the success criterion is just partially checked. Other users were more specific in identifying such cases: when a tool is not able to access URLs that are protected by login; when tools are not able to perform their assessment when specific technologies are considered or when dynamic pages are considered; when issues are in the content of the page, rather than in its structure; when checking colour contrast on pseudo-elements; when they have to ~~analyze~~analyse mobile apps, when they have to ~~analyze~~analyse different types of documents/formats (i.e. svg, pdf), and other specific situations (e.g. shadow DOMs, content inside frames). Also, one user mentioned the inability of tools to perfectly emulate a braille reader; another user mentioned the inability for tools to evaluate how properly images and alternative texts are used in a website. One of the users mentioned that tools are not able to cover all

WCAG success criteria, and it would be useful to know the rules (testing algorithms) used and which published ACT rules are covered. Another theme regarded limitations concerning the **preferences and parameters that users can specify for the analysis** (8 users mentioned this point). Some users mentioned as a limitation the number of pages to consider for the evaluation, one user mentioned the depth of the analysis. One user highlighted as a limitation the lack of compliance with specific standards. One user highlighted that there are some tools which are not very up-to-date. Another comment indicated that the versions of the various languages (e.g. JavaScript) and frameworks (e.g. Bootstrap) that the tool is able to address could represent a limitation, and thus it should be clearly indicated. Another type of limitation regarded the occurrence of **situations in which the results can be wrong (i.e. false positive or false negatives), or ambiguous**. Seven users mentioned that tools should clearly indicate situations that can generate false positives/negatives in the evaluation or indicate when the evaluation could generate multiple interpretations. In this regard, one user mentioned that the ARC Toolkit issues warning for cases that may or may not be a problem depending on the context. Another theme regarded **the lack of a clear declaration highlighting the need for manual checks (with possible guidance).** Five users emphasized the fact that an automatic validation is never complete and exhaustive, therefore tools should clearly highlight this, also possibly providing guidance for carrying out manual checks. In addition, one user said: "*An automatic evaluation is not enough to guarantee that a site is accessible. Heads up for manual checks would be appreciated; some tools do that*." Finally, as **further aspects** mentioned, one user suggested that it would be useful to know in advance the behaviour of the tool compared to a benchmark (in terms of false positive, false negatives, coverage), acknowledging that this would require to have a 'normalized' corpus and process to assess evaluation tools. One user highlighted that tools should be more explicit about their pricing options. One user would like to have more information about situations in which different tools return different results. One user mentioned that tools should mention the possible improvements. Another aspect mentioned by a user is that sometimes tools are too "code-based".

Regarding the users' ratings, as it can be seen from Table 4, since the median (Mdn) values are higher than mean (M) values, the data distribution is more concentrated on the right-side, corresponding to the higher scores. In addition, while the range (which gives a measurement of how spread out the entire data set is) is high (Min=1, and Max=5), the interquartile range (which gives the range of the middle half of a data set) is low (IQR=1), which means that the middle half of the data shows little variability.

5.4 Effect of Frequency of Use of Tools and Level of Technicality of Users on Ratings to S1-S6 Aspects

We were also interested in understanding whether there was some effect of users' technicality level and/or familiarity of users with tools on the ratings given to S1-S6 statements.

On the one hand, users were divided into 'non-technical' people (i.e., people who should refer to a technical expert to solve accessibility problems emerged from automatic validation, thereby people with few or no skills in developing or writing code, such as Web commissioners, project managers, legal workers, UX/UI designers) and 'technical' people (i.e., people able to solve a problem of accessibility that has emerged from the automated validation at software implementation level, thereby people with programming skills). On the other hand, regarding the familiarity of users with accessibility validation tools, we rated it in terms of how many times the users declared to use such tools for their work. Since there were 48 users who declared not to use tools, we rated them as unfamiliar/infrequent users (frequency of use of tools=0). As for the other 90, we computed in the 'non-frequent' group of users those who use tools either once a year (frequency of use of tools=1) or once a month (frequency of use of tools=2), whereas those who use tools either once a week (frequency of use of tools=3) or once a day (frequency of use of tools=4) were considered as 'frequent' users. To sum up, in the end, we had 30 users in the 'non-technical' group and 108 in the 'technical' group; 43 users in the 'frequent' group and 95 in the 'non-frequent' group.

To understand whether there was some significant difference between the technical vs non-technical group and the expert and non-expert group in the scores given to the various aspects S1-S6, we first checked the normality of the concerned distributions. Since all the distributions were non-normal, we ran the non-parametric unpaired two-sample Wilcoxon test, finding a significant effect of the frequency of use of the tools to some of the statements (see Table 5). In particular:

- *Effect of frequency of use of tools on the usefulness of having that the tool specifies how it categorizes evaluation results (errors, warnings, etc.).* The frequency of use of tools has an effect on aspect S2. In particular, the unpaired two-samples two-tails Wilcoxon rank sum test highlighted a significant difference (p = 0.004958) of the median value of the scores to this question between the frequent users and the non-frequent users. When testing whether the median of the 'frequent' group was greater than the median of the 'non-frequent' group, the Wilcoxon one-tail test suggested (p= 0.002479) that those who use more frequently the tools see more useful that the tool specifies how it categorizes the results, compared to those who use less the tools. This seems to indicate that those who use more the tools are aware of the importance of this aspect in the work of understanding accessibility problems and correct them.
- *Effect of frequency of use of tools on the usefulness of having the tool provide general measures that make explicit the level of accessibility of the website/mobile app.* The frequency of use of tools has an effect on aspect S3. In particular, the unpaired two-samples two-tails Wilcoxon rank sum test highlighted a significant difference (p= 0.01834) of the median value of the scores to this question between the frequent users and the non-frequent users. When testing whether the median of the 'infrequent' group was greater than the median of the 'frequent' group, the Wilcoxon one-tail test suggested (p= 0.009171) that those who use less frequently the tools see more useful that the tool is able to provide general measures, compared to those who use the tools more frequently. This could be explained with the fact that those who use the tools more frequently rely less on such 'summative' accessibility measures, which typically are more directed to less skilled users who are often more interested in understanding whether the Web application is accessible but are not particularly involved in the work of correcting accessibility errors.
- *Effect of frequency of use of tools on the usefulness of having that the tool gives indications on its limitations.* The frequency of use of tools has an effect on question S6. In particular, the unpaired two-samples two-tails Wilcoxon rank sum test highlighted a significant difference (p= 0.00148) of the median value of the scores to this question between the frequent users and the non-frequent users. When testing whether the median of the 'frequent' group was greater than the median of the 'infrequent' group, the Wilcoxon one-tail test suggested (p= 0.0007402) that those who use more frequently the tools see more useful that the tool gives indications on its limitations, compared to those who use the tools less frequently. This could be explained by the fact that those who use tools more frequently, and thus are more involved in the work of correcting accessibility problems, see as more useful having an explicit indication of limitations of the tools (an aspect that tools tend not to emphasize much), to better understand the actual abilities of the tools in performing the validation.

Stat.	FREQ (M)	NON-FREQ (M)	Shapiro-Wilk normal. test H0= normal distrib. (signific. level= 0.05)	FREQ (Mdn)	NON-FREQ (Mdn)	Wilcoxon rank sum test with continuity correction (two-sided)
S1	4.67	4.6	S-W (freq) p-value = 5.03e-11 S-W (nonfreq) p-value = 5.62e-14	5	5	p-value = 0.3099
S2	4.74	4.44	S-W (freq) p-value = 5.622e-12 S-W (nonfreq) p-value = 1.903e-12	5	5	p-value =0.004958* Wilcoxon (one-sided): p-value = 0.002479*
S3	3.93	4.43	S-W (freq) p-value = 1.165e-05 S-W (nonfreq) p-value = 6.766e-12	4	5	p-value =0.01834* Wilcoxon (one-sided): p-value = 0.009171*
S4	4.37	4.44	S-W (freq) p-value = 1.793e-09 S-W (nonfreq) p-value= 1.516e-12	5	5	p-value =0.4798
S5	4.56	4.73	S-W (freq) p-value = 4.059e-10 S-W (nonfreq) p-value= 2.2e-16	5	5	p-value = 0.3351
S6	4.56	4.06	S-W (freq) p-value = 8.042e-10 S-W (nonfreq) p-value = 2.004e-09	5	4	p-value = 0.00148* Wilcoxon one-sided: p-value = 0.0007402*

Table 5: Statistical analysis of the data based on use of accessibility validation tools

We found no significant effect of the level of technicality to all the above statements, as it is possible to see from the below Table 6.

Stat.	TECH (M)	NON-TECH (M)	Shapiro-Wilk normality test H0= normal distrib. (signific. level= 0.05)	TECH (Mdn)	NON-TECH (Mdn)	Wilcoxon rank sum test with continuity correction (two-sided)
S1	4.61	4.67	S-W (tech) p-value=1.278e-15 S-W (nontech) p-value = 1.418e-07	5	5	p-value =0.9557
S2	4.56	4.43	S-W (tech) p-value = 7e-15 S-W (nontech) p-value = 5.842e-07	5	5	p-value =0.4923
S3	4.21	4.47	S-W (tech) p-value = 1.186e-11 S-W(nontech) p-value=1.701e-06	4	5	p-value =0.1758
S4	4.4	4.47	S-W(tech) p-value = 3.387e-14 S-W(nontech) p-value=1.701e-06	5	5	p-value =0.959
S5	4.66	4.7	S-W (tech) p-value = 2.2e-16 S-W(nontech) p-value=5.2e-09	5	5	p-value =0.8465
S6	4.26	4.03	S-W(tech) p-value=2.37e-12 S-W(nontech) p-value =0.000144	5	4	p-value =0.2103

Table 6: Statistical analysis of the data based on user technical level

6 THE USER TEST

In order to have more direct user feedback about tools' transparency, a user test was carried out. The main objective of the test was to see how the transparency criteria identified in Section 3 were assessed by some users who carried out specific tasks relevant for transparency aspects by using three different accessibility validators, and thus were able to provide more focused comments on how they support the proposed criteria.

For this purpose, we selected three tools (MAUVE++, QualWeb, and Lighthouse) that seem sufficiently representative since they are public and provide updated support even for most recent accessibility guidelines. In addition, they are available in different ways: one is mainly a stand-alone Web application (MAUVE++ [Broccia et al., 2020]), one (Lighthouse) is integrated in a widely used browser (Chrome), and one is open source (QualWeb). Lighthouse was not considered in the initial comparative analysis because it was not listed in the W3C tool list from which we selected the analysed tools. In the test, the users had to perform some tasks using the three different accessibility validation tools, and then answer some related questions.

6.1 Participants and Tasks

The test was carried out by eighteen users (13 Males, 5 Females), the age ranged between 25 and 57 (mean 41.27). Regarding their role, 5 indicated accessibility experts, 7 Web developers, 3 Web commissioners, 3 Web developers and accessibility experts. In terms of familiarity with the tools, we considered a scale from 1 (none) to 5 (full), the average was 2.83 for MAUVE++, 1.05 for QualWeb, 1.55 for Lighthouse. The tests were performed remotely with the support of videoconference systems, and they were video-recorded with the users' permission. During the test, there were always two moderators that, after introducing the study's motivations, followed the execution of the test (users were asked to share their screen with them). During the test, the moderators did not intervene to help users to perform the tasks, even when their answers were not correct. The order of the tools was counterbalanced in order to control the potential confounds created by order/learning effects on task performance. The users accessed the versions of the tools available on the Web in July 2021.

Before starting the test, the moderators briefly introduced the concept of transparency and the goal of the test. Then, they provided users with the URL of the three tools, the link to the page to evaluate using them (https://en.wikipedia.org/wiki/Main_Page/), and the link to the questionnaire. Since the users had to fill in the questionnaire while performing the tasks of the test, to prevent any influence on their questionnaire responses, at the beginning of the test they were asked to open the questionnaire page on a part of the screen that was not shared with the moderators. The duration of each test session ranged between 40 and 70 minutes.

For each of the three tools, the users were asked to do two tasks and then answer some associated questions. Such tasks have been defined in order to drive the users to test the different ways to evaluate and produce the validation results in all the tools. In particular, the tasks were identified in such a way to make users first explore the main functionality provided by each tool, then use the tool to do a concrete accessibility validation on a page and look for specific information that the tool provides to the user during the validation (to be able to assess tool's transparency-related aspects in a more informed manner); then, they had to answer to some associated questions. The sequence of tasks and questions was provided to users via a single Google Form document. In that document, while some questions were aimed to gather subjective user's feedback (i.e. tool's aspects that users appreciated more, from a transparency-related perspective), most of the questions were aimed at gathering information to more objectively understand whether users were able to find –when available– and also understand specific transparency-related information provided by the tool, according to the transparency criteria C1-C5 identified before.

The sequence of tasks and questions included in the document is detailed below. Please note that, for better clarity, at the end of the questions that address specific transparency criteria, we added the reference to the relevant criterion.

Task1. Access the tool, browse the information it provides regarding its own functionality, then answer:

- *Q1: Does the tool state at some point which standards (e.g., WCAG 2.1, WCAG 2.0, EN 301 549, etc.), success criteria and techniques it supports? Possible options are: Standards, Success Criteria, Techniques, Descriptions of the accessibility aspects supported, No information. Then, please indicate the supported standards.* (Question related to criterion C1)

Task 2. Using the tool, validate the page: https://en.wikipedia.org/wiki/Main_Page/, browse the results, then answer the following questions:

- *Q2: How are the results of the assessment of each Web page element classified? Please specify the categories used by the tool.* (Question related to criterion C2)
- *Q3: Did the tool provide you with information explaining the meaning of each category (please also copy-paste the relevant information found)?* (Question related to criterion C2)
- *Q4: How many errors have you found?* (Question related to criterion C2)
- *Q5: How many elements does the tool indicate that it is not able to check automatically*? (Question related to criterion C2)
- *Q6: Does the tool offer numerical indicators that provide a measure of the level of accessibility of the Web page and/or the completeness of the evaluation?* (Question related to criterion C3)
- *Q7: What do these measures describe in your opinion?* (Question related to criterion C3)
- *Q8: Do you find the presentation of the evaluation results useful and understandable? Please motivate the answer.* (Question related to criterion C3)
- *Q9: ~~Analyze~~Analyse one of the errors the tool detects. What information is given on the type of problem?* (Question related to criterion C3)
- *Q10: What information does the tool provide on how to solve the problems identified in the assessment?* (Question related to criterion C4)
- *Q11: Does this information seem sufficient to you to solve the problem? Explain your answer* (Question related to criterion C4)
- *Q12: Browsing the tool, do you find a point where the tool states its limitations in carrying out the accessibility assessment?* (Question related to criterion C5)

Then, some final considerations on the tool were collected by users, who had to answer the following questions:

- *Q13: In terms of transparency, which features of this tool did you like the most?*
- *Q14: In terms of transparency, what aspects of this tool did you dislike?*
- *Q15: In terms of transparency, which features should this tool have that you have not found?*

Lastly, the users had to rate on a scale from 1 (non-transparent) to 5 (fully transparent) the three tools.

6.2 MAUVE++

Regarding whether MAUVE++ provides information on the guidelines supported (Q1), 11 answered correctly, 7 answered partially (4 users found only standards, 3 did not find the techniques). As for how the tool classifies the results of assessing each Web page element (Q2) and whether it provides information on such classification (Q3), 11 users correctly recognised the classification by indicating the three types of results provided by the tool (error, warning, success), 2 users partially recognised it, 5 users were not able to recognise it and indicated other types of information generated by the tool. When asked about the number of errors (Q4) and warnings (Q5) provided by the tool as a result of the evaluation, 4 users correctly answered these questions, showing that they understood the difference between errors and warnings, while the remaining 14 provided an answer that was just partially correct: most of the time they did not understand the category of "warnings". This was partly due to the fact that, in question Q5, we did not explicitly ask to count the "warnings", but we asked more generically to find the elements that cannot be automatically evaluated by the tool. Therefore, it seems that some users did not clearly understand what "warnings" actually mean.

Regarding whether the tool provides quantitative indicators of the level of accessibility of the considered page overall and the completeness of the evaluation (Q6), the answers were all correct: all users seem aware that MAUVE++ does indeed provide such information. Regarding whether the presentation of the results is useful and comprehensible (Q8), the answers were yes (12 users), in part (5), no (1). The positive comments regarded the various ways to present the evaluation results, along with relevant detailed information. Those who were partially positive found that some information was not immediately understandable. The negative user complained about problems in the generated PDF report.

Then users were asked to analyse one of the errors detected, indicate the associated information provided by the tool, and also assess whether it was sufficient to solve the problem. Sixteen users provided the information that the tool shows as associated with a specific error, while two users were not able to provide it. In addition, 14 users correctly provided the information that the tool shows about how to solve a specific accessibility issue, whereas 4 users were not able. In addition, when asked to assess that information (question Q11), 11 users judged it as sufficient, 3 partially sufficient, and 4 not sufficient: in particular, the need for simple and clear examples of solutions was highlighted.

Regarding whether the tool provides clear information on its limitations (Q12), there were mixed answers: 9 positive and 9 negative. The positive ones indicated various types of information provided by the tool, such as a pie chart showing the percentage of the checkpoints actually assessed (evaluation completeness), and a file describing the supported success criteria.

The question regarding what aspects they liked most in the transparency perspective (Q13) received several answers, with varied relevance: the possibility to view the validation results from different viewpoints (end user and developer) with the possibility to filter the results, the possibility to associate an error to its position in the code with one click, the indication of the techniques and criteria actually checked by the tool, the indication of how the tool performs the validation (with the support of an XML specification of the guidelines), the interactive preview with the possibility to expand with the occurrences of each type of error, the overall accessibility scores.

Regarding the aspects that they did not like (Q14), the answers were varied as well: sometimes the error description was found unclear, whereas a comment highlighted as a drawback that the internal specification of the guidelines was not public. In addition, how the tool supports users in finding the points having errors was judged improvable, as well as the possibility to add further filters in the presentation of the results for developers. Also the difference between errors and warnings was found not to be clearly explained. Regarding further aspects to improve, example points that were mentioned were: greater clarity in the description of the criteria, the possibility to have clickable elements in the evaluation summary, the need to better highlight that warnings were points in which a further check by the user is needed.

6.3 QUALWEB

As for the question whether the tool provides information on the guidelines supported (Q1), 7 users answered correctly, 9 answered partially (only the standards were found), and 2 answered incorrectly; most users noticed that QualWeb also provides support for ACT rules. Regarding the categorization of accessibility results (Q2, Q3), 16 users correctly recognised it (errors, warnings, passed, not applicable), 1 user did not find the categorisation at all, 1 user provided a partially correct answer.

When asked about the number of errors detected (Q4) and warnings (Q5), 9 users correctly understood the distinction between errors and warnings, 8 users just partially understood it (most of the time they did not recognize the warnings); 1 person did not understand the categorization at all.

Regarding whether the tool provides overall accessibility indicators (Q6) and their explanations (Q7), 9 answered positively and 9 negatively. Those who were positive indicated the number of errors and checks positively passed as such indicators.

The question (Q8) about whether the results presentation is useful and comprehensible, received 12 "partly" answers, 5 "yes" and 1 "no". Among the negative aspects, some users mentioned the lack of display of the errors directly in the Web page user interface and code, the lack of graphical representations of the results, the fact that results are shown without following a clear order, and that a browsable overall view was missing. In contrast, on the positive side, they mentioned the summary of the assessment results, the filters on the results shown, the possibility to see the code extract associated with the error, the fact that sometimes the tool provides useful information to analyse the errors, the use of the accordion widget to provide more detail on the errors detected.

Then, users were asked to analyse one of the errors and report the associated information provided by the tool (Q9). Regarding the information on the problem detected, all 18 users replied correctly: they indicated that the tool provides information on the problem type, the associated technique and the excerpt with the page element involved, along with access to the W3C relevant information.

Users also had to find the information that the tool shows for solving a specific issue (Q10): 14 users were able to find it, whereas 4 users did not. As for whether the provided information was judged sufficient (Q11), ten answered positively, while eight expressed concerns, such as that the information is suitable only for developers or accessibility experts, and the need to better understand how the problem should be solved or to explicitly indicate the most important aspects to correct or to show the error in the Web page code.

Regarding whether the tool indicates its limitations (Q12), 15 answered negatively, and the remaining three correctly reported that they found some indications i.e. in the referred W3C documentation, in the indication of the guidelines supported, and the non-applicable techniques. For the aspects they liked most from a transparency perspective (Q13), users mentioned the information about the number of inapplicable assessments and the number of those actually performed, the error classification, and that for each error the violated success criteria and a possible solution are indicated, the fact that it is possible to filter the results' presentation, that it provides the excerpt of the code corresponding to the error, and that it is open source.

Concerning the aspects that they did not like (Q14), the users mentioned, in the user interface, the lack of : display of the elements that generated an accessibility error, any order in the result list, a dashboard with a graphical summary of the results, concrete indications on how to solve the problems, and the limited support in identifying the error in the source code.

They indicated as desired features (Q15) a preview of the page with the errors highlighted in it, the organization of the errors according to the WCAG principles, more concrete information on how to solve the problems, some graphical summary of the results, more information on ACT rules, better explanation of the coverage of WCAG guidelines.

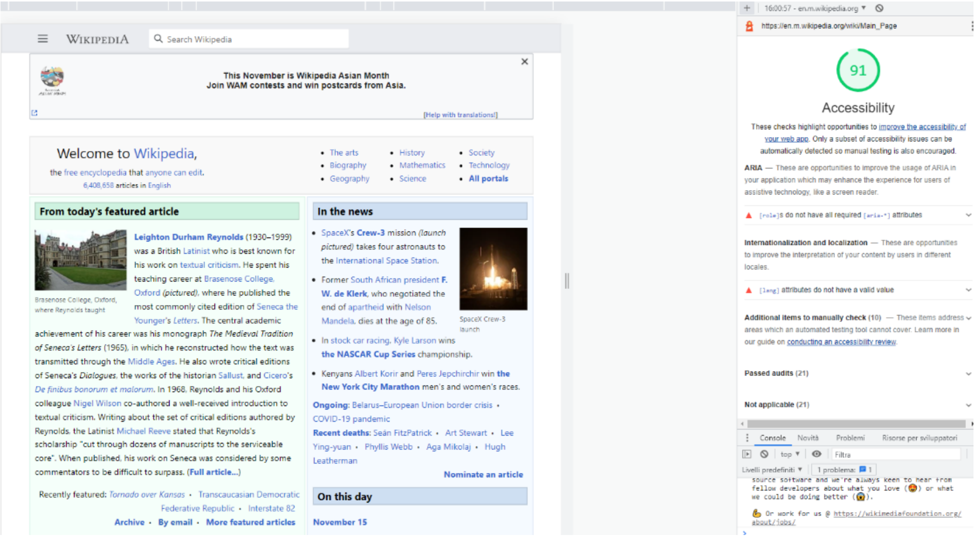
6.4 LIGHTHOUSE

6.4.1 Description of Lighthouse

Lighthouse (see Figure 1) is an open-source, automated tool for improving the quality of web pages. It has audits for performance, accessibility, progressive web apps, SEO and more. It can be set up and used in a variety of ways, which makes it a good option for both technical and non-technical users. In the test we used the automatically available version in Chrome, with no setup or extensions to install, by opening Chrome Developer Tools panel and then choosing the Lighthouse tab at the top.. One disadvantage of using Lighthouse in the browser is that only one page can be audited at a time, which makes the process cumbersome for sites with a large number of pages.

The Lighthouse tab in Chrome DevTools has a straightforward interface which allows users to choose to run just an accessibility audit or run additional audits (including e.g. Performance, Progressive Web App), and it also allows the user to choose between running the audit on desktop or an emulated mobile device. Once those choices have been made, the user has to press the "Generate report" button to run the selected audit(s) directly in the browser. Lighthouse runs its audits against the currently-focused page, then opens up a new tab directly showing a HTML-based report of the results. The report can also be saved in JSON, or it can be opened in the online Lighthouse Viewer (by submitting the JSON output of a Lighthouse report); in addition, users can also share their reports by creating secret GitHub Gists (the benefit of Gists is free version control), which can also be viewed in the viewer.

A feature of The Lighthouse accessibility audit report is a metric called *Accessibility Score,* which is generally shown at the top of the report. This score is a weighted average of all of the accessibility 'audits' Lighthouse runs on the current page, and it is based on Deque's axe user impact assessments (which Deque uses for its own axe browser's extension), which cover WCAG 2.0 and WCAG 2.1 for A and AA conformance levels. However, it is worth pointing out that within Lighthouse, the end user cannot directly find an explicit reference to the WCAG guidelines: the fact that Lighthouse covers WCAG 2.1 (A, AA) has to be implicitly deduced from the fact that it implements the axe-core rules (as reported in https://web.dev/accessibility-scoring/) which is a JavaScript library (used in the axe browser's extension) that covers WCAG 2.1 (A and AA). So, from this point of view, Lighthouse is not very transparent. Also, Lighthouse does not let the user know which WCAG rules are being violated. In addition, despite being based on axe's list, Lighthouse actually misses some issues that the axe extension is able to catch (i.e. it does not implement the full list of axe-core rules), so inconsistent results can be generated between axe and Lighthouse, which might confound users.



This November is Wikipedia Asian Month
Join WAM contests and win postcards from Asia.

Welcome to Wikipedia,
the free encyclopedia that anyone can edit.
6,408,658 articles in English

From today's featured article

Brasenose College, Oxford, where Reynolds taught

Leighton Durham Reynolds (1930–1999) was a British Latinist who is best known for his work on textual criticism. He spent his teaching career at Brasenose College, Oxford *(pictured)*, where he published the most commonly cited edition of Seneca the Younger's *Letters*. The central academic achievement of his career was his monograph *The Medieval Tradition of Seneca's Letters* (1965), in which he reconstructed how the text was transmitted through the Middle Ages. He also wrote critical editions of Seneca's *Dialogues*, the works of the historian Sallust, and Cicero's *De finibus bonorum et malorum*. In 1968, Reynolds and his Oxford colleague Nigel Wilson co-authored a well-received introduction to textual criticism. Writing about the set of critical editions authored by Reynolds, the Latinist Michael Reeve stated that Reynolds's scholarship "cut through dozens of manuscripts to the serviceable core". When published, his work on Seneca was considered by some commentators to be difficult to surpass. (**Full article...**)

Recently featured: *Tornado over Kansas* · Transcaucasian Democratic Federative Republic · Interstate 82
Archive · **By email** · **More featured articles**

In the news

- SpaceX's **Crew-3** mission *(launch pictured)* takes four astronauts to the International Space Station.
- Former South African president **F. W. de Klerk**, who negotiated the end of apartheid with Nelson Mandela, dies at the age of 85.
- In stock car racing, Kyle Larson wins **the NASCAR Cup Series** championship.
- Kenyans Albert Korir and Peres Jepchirchir win **the New York City Marathon** men's and women's races.

SpaceX Crew-3 launch

Ongoing: Belarus–European Union border crisis · COVID-19 pandemic
Recent deaths: Seán FitzPatrick · Art Stewart · Lee Ying-yuan · Phyllis Webb · Aga Mikolaj · Hugh Leatherman
Nominate an article

On this day

November 15

Accessibility

These checks highlight opportunities to improve the accessibility of your web app. Only a subset of accessibility issues can be automatically detected so manual testing is also encouraged.

ARIA — These are opportunities to improve the usage of ARIA in your application which may enhance the experience for users of assistive technology, like a screen reader.

▲ `[role]`s do not have all required `[aria-*]` attributes

Internationalization and localization — These are opportunities to improve the interpretation of your content by users in different locales.

▲ `[lang]` attributes do not have a valid value

Additional items to manually check (10) — These items address areas which an automated testing tool cannot cover. Learn more in our guide on conducting an accessibility review.

Passed audits (21)

Not applicable (21)

```
source software and we're always keen to hear from
fellow developers about what you love (😍) or what
we could be doing better (😱).

💪 Or work for us @ https://wikimediafoundation.org/
about/jobs/
>
```

Figure 1: The Lighthouse tool

Lighthouse uses a simple 1-100 grading system for auditing pages, further divided into 3 sub-intervals associated with different colours: red colour applies to scores between 0 and 49, which are considered poor, 50–89 is associated with a yellow colour (needs improvements), 90–100 scores are associated with a green colour (good). It also shows where improvements can be made. However, it is worth emphasising that a score of 100 does not mean that the site is perfectly accessible, as Lighthouse uses the axe-core accessibility testing library with a custom (namely: partial) set of rules for its tests, and of course manual testing is still needed.

As mentioned before, the Lighthouse Accessibility score is a weighted average of the accessibility factors (or 'audits') considered by Lighthouse, and their weighting is based on axe user impact assessments (manual audits are of course not included in the score calculation). Each 'accessibility audit' is pass or fail, which means that a page does not get points for *partially* passing an accessibility audit. For example, if half the buttons have screen-reader friendly names, and half do not, you will not get "half" of the weighted average: you get a 0 because that accessibility criterion needs to be implemented correctly throughout the page.

. As for the results provided by the tool, they can be grouped into: Failing Elements, Additional items to manually check, Passed audits, Not applicable. The Failing elements are then further categorized into the accessibility criteria considered by Lighthouse (Navigation, ARIA, Names and Labels, Contrast, Tables and Lists, Best Practices, Audio and Video, and Internationalization and Localization).

For each violation found by Lighthouse, the tool provides: a short description of the issue, a "Learn more" link to the page on the web.dev website explaining the issue (thus, Lighthouse does not let the user know exactly which WCAG rules are being violated), and a list of the elements in the web page that fail that particular accessibility criterion (some of such elements are accompanied with a screenshot of the involved element shown within the page, so the user can visually locate it). In addition, it is also possible to see where the violation occurred in the HTML code.

Finally, as for its limitations, Lighthouse does not explicitly provide any indication about its limitations, apart from a general statement saying that only a subset of accessibility issues can be automatically detected, thereby manual testing is encouraged. Being available also as a browser extension (in Chrome and Firefox), it is able to validate dynamic content.

6.4.2 Answers to the questions regarding Lighthouse by test participants

Regarding whether Lighthouse (Figure 1) provides information on the guidelines supported (Q1), 16 users (correctly) answered that the tool does not provide such info, 2 answered positively, even though only to some extent: both of them mentioned a limitation connected with checking ARIA standards, whose support is mentioned only after validating a specific page.

As for how the tool classifies the results of the assessment and whether it provides information on such classification (Q2, Q3), nine users correctly indicated the different types of results, which in Lighthouse are: ARIA, Navigation, Internationalization and localization, Additional items to manually check, Passed audits, Not applicable. It is worth noting that in Lighthouse errors do not have a category per se, but they are considered in the aforementioned categories, depending on the type of problem found. Four users recognized the categories just in a partial manner, while five users did not recognize the different ways in which the tool classifies the results at all.

When asked how many errors the tool identified in the considered Web page (Q4) and how many elements the tool indicated that it could not automatically check (Q5), i.e. the so-called ‘warnings’ or ‘cannotell’, 6 users correctly identified both the numbers of errors and warnings, 5 users did not identify them at all, 7 users identified them just in a partial manner.

Regarding whether the tool provides quantitative indicators of the accessibility and completeness of the evaluation (Q6), all participants answered affirmatively (and correctly), also saying that such indicators offer a succinct summary measure of the accessibility of the considered page. However, a few of them expressed some doubts about how they were concretely calculated.

As for the question on whether the results’ presentation is useful and comprehensible (Q8), the answers were yes (N=4), in part (N=7), no (N=7). The positive comments said that the presentation is quite immediate and easy to understand, as it presents a list of errors clearly differentiated into a number of macro-categories (passed, not applicable, fail), and it also offers the possibility to automatically refer to the relevant code portions (by clicking on the associated link). Those who were partially positive found that sometimes it was not easy to understand which part(s) of the page generate(s) errors; another user, while acknowledging that the tool highlights errors, it does not specify in what way the checks fail; another user complained that it was not easy to find immediately the total number of errors; another user highlighted the need to group the errors according to their type; another user said that sometimes the explanations are extremely short. Finally, a user suggested including an explicit reference to the accessibility standard(s) referred by the tool. The majority of the negative comments regarded the visualization of the results, which was found unclear, with a confusing layout, and more in general too concise, also lacking structured visualizations that include graphs. Three users complained about the fact that the tool does not allow users to understand where the errors/warnings are located within the page. One user complained that the visualization of results does not help in concretely solving the accessibility issues found.

Then the participants were asked to ~~analyze~~analyse one of the errors detected and indicate the associated information provided by the tool (Q9). The majority (N=16) correctly found in the tool the information about the errors detected, two users did not. In addition, 14 users correctly provided the information that the tool offers to solve the found issue (Q10), while 4 users did not. Regarding whether such provided information was perceived as sufficient to solve the problem (Q11), 7 users found that the tool does not provide sufficient information. Seven users were overall satisfied about this aspect. The remaining 4 users expressed a ‘borderline’ point of view: while acknowledging that the tool provides some information, this was found sufficient to address just the simplest cases, while in other cases further detail is needed, especially by unskilled users.

Regarding whether the tool provides clear information on its limitations (Q12), 11 users incorrectly answered yes, the remaining 7 answered no: among them, 4 users explicitly declared not to have found it, the remaining 3 complained that this information is provided only in an implicit manner and further clarity should be added.

Among the aspects that users appreciated most (Q13), they mentioned the fact that the tool indicates what it cannot check, and emphasizes that additional manual tests should be done on specific elements (N=3), the result presentation is immediate, concise and overall easy to use (N=3) and also uses icons to categorize the type of results, the links available for further inspection (N=3), the categorization of the errors according to the various concerned aspects (N=2), the visualization of the global level of accessibility (N=2), the fact that it lists the items that passed the automatic check, those that need further manual check, and those that are non-applicable.

Among the aspects that were not liked (Q14) the answers were varied as well. Three users complained about issues connected with standards: in particular, one complained that the tool refers only to ARIA standard, one complained about the lack of clarity on the specific standard referred, and another user declared to have not found any possibility to set the conformance level ("A", "AA", "AAA"). Three users complained about aspects related to problem resolution, in particular the need to give more information about how to concretely solve the errors**.** Moreover, aspects related to errors were mentioned by users. In particular, two users said that it was not clear which elements generate errors, nor were the referred criteria clear. Another user complained about the fact that Lighthouse does not clearly indicate the errors and warnings and also the associated descriptions are a bit difficult to understand. Another user highlighted the inability to have the results of the analysis shown directly in the page and within the code. Five users raised concerns associated with the usability, intuitiveness and the clarity of the tool and the provided visualizations. A couple of users pointed out that the presentation of results done with the accordion-like widgets was very minimal. Another user noted that if a user exploits a small monitor, it could be difficult to see the long list of results that the tool often provides, especially because the top part of the UI is occupied by the preview; three users raised concerns associated with the limitations of the validation, which they found unclear.

Regarding the additional features/characteristics that users would have liked to find in the tool (Q15), the participants mentioned:

- a more detailed description of the various errors and further information for solving them;
- add also the warnings to the categorization,
- make the references to standards and guidelines explicit,
- add further graphical visualizations,
- add an indication of the settings according to which the analysis has been done,
- make a more ‘global’ view of the code available,
- add further filters for performing a validation,
- make clearer and possibly group together in a new section the errors that should be checked manually,
- add further references to the code ~~analyzed~~analysed,
- add further information on the tool’s limitations.

6.5 Comparative Analysis of the Answers Associated with Transparency-Related Criteria

Table 7 provides a summary comparison of the three tools on the questions in which users were required to find specific information about a tool; for the other questions, namely those which asked users for a subjective opinion, such as the aspects they liked most/least, the reader can still refer to previous sections 6.2-6.4.

Since these questions were identified to check to what extent users are able to find specific, transparency-related information of each tool (according to the criteria C1-C5), in the first column of each row, within brackets, we report the associated transparency criterion. The other three columns (one for each tool) specify the number of users who replied *fully/partially/not at all correctly* to these questions. It is worth pointing out that sometimes we used more than one question to cover one single transparency criterion: for instance, questions Q4 and Q5 both aim to cover criterion C2, to see if the user actually understands the distinction between errors (Q4), and warnings (Q5).

Questions (Criteria)	**Information requested in the test**		**MAUVE ++ # users**	**QualWeb # users**	**Lighthouse # users**
Q1 (C1)	Info about standard, success criteria, techniques supported by the tool was (Partially/Fully/Not at all) correctly found by X users	Partially	7	**9**	0
		Fully	11	7	**16**
		Not at all	0	**2**	**2**
Q2, Q3 (C2)	The categorisation of accessibility issues into different types, as it is provided by the tool, was (P/F/N) correctly recognised by X users	Partially	2	1	**4**
		Fully	11	**16**	9
		Not at all	**5**	1	**5**
Q4, Q5 (C2)	The categorisation of accessibility issues into different types, as it is provided by the tool, has been (P/F/N) correctly understood by X users	Partially	**14**	8	7
		Fully	4	**9**	6
		Not at all	0	1	**5**
Q6, Q7 (C3)	Summative indicators of accessibility level, as provided by the tool, have been (P/F/N) correctly found by X users	Partially	0	0	0
		Fully	**18**	9	**18**
		Not at all	0	**9**	0
Q9 (C3)	The information about a specific accessibility issue, as provided by the tool, has been (P/F/N) correctly found by X users	Partially	0	0	0
		Fully	16	**18**	16
		Not at all	**2**	0	**2**
Q10 (C4)	The information about how to solve a specific accessibility issue, as provided by the tool, was (P/F/N) correctly found by X users	Partially	0	0	0
		Fully	**14**	**14**	**14**
		Not at all	**4**	**4**	**4**
Q12 (C5)	X users correctly/incorrectly answered the question whether the tool provides specific information about its validation limitations	Correct	**9**	3	7
		Incorrect	9	**15**	11

Table 7: A summary of the answers that users provided in the test, in reply to questions asking specific transparency - related information about each tool

As for the provision of the guidelines, standard and success criteria (C1), the majority of users correctly recognised that Lighthouse did not provide such information in an explicit manner. As for the categorisation of issues (C2), QualWeb was the tool that received the highest number of correct answers from our users when they had to indicate the classification used by that tool. However, the majority of the participants did not prove to completely understand the difference between warnings and errors. All the users were able to correctly find the indicators provided by MAUVE++ and Lighthouse (C3), and the information provided by the three tools about a specific issue was correctly found by the majority of users (MAUVE++ and Lighthouse), or all users (QualWeb). Also the information about how to solve a specific issue (C4) was correctly found by a good number of users for all the tools, N=14). Finally, it seems that the information about the limitations was not very evident in any of the three tools (apart from MAUVE++, where half participants correctly found this information and half did not, for the other two tools the majority of answers were incorrect).

Lastly the users were asked to provide an overall rating of the transparency of each tool on a scale from 1 (no transparency) to 5 (fully transparent). The results were MAUVE++ (min: 1, max: 5, mean: 3.88, median: 4), QualWeb (min: 1, max: 4, mean: 3, median: 3), Lighthouse (min: 1, max: 4, mean: 2.44, median: 2). Overall, it seems that as of yet no one of the tools fully supports transparency. In the case of MAUVE++ it was appreciated the possibility of having multiple views on the errors detected (summary tables and annotated source code). In QualWeb the summary tables were appreciated as well. Lighthouse does not yet seem to provide various relevant pieces of information in a clear manner.

We aimed to compare the means to see if they were statistically significant. We could not apply the one-way repeated measures ANOVA because the three distributions were not normal. However, since the variances were found to be homogeneous (we apply the Levene test, p-value= 0.6775, H0: the variances of the three groups are homogeneous), and there were more than two groups to compare, we applied the Kruskal-Wallis test, which yielded a p-value=0.0001, thereby indicating that there were significant differences between the three groups. However, since the output of the Kruskal-Wallis test indicates that there is a significant difference between groups, but not between which pairs of groups, we used a Wilcoxon pairwise comparison between group levels with Bonferroni correction, which showed that the mean scores associated respectively with MAUVE++ and QualWeb, and MAUVE++ and Lighthouse are significantly different ($p < 0.05$), which generally implies that these results are also valid for the general population (not just for the sample considered for the test).

7 DISCUSSION

Based on the answers provided in the survey, the user test, and our analysis of the state of art we can identify a number of general aspects that are important in order to better address transparency, and to promote trust of such tools on the part of users.

People need different information on and representations of the validation results depending on their use of the tools. The survey clearly indicated that users' expected information regarding transparency depends on the frequency of their use, which in turn generally depends on their role. Those who access the validation tools less frequently, mainly to check the level of accessibility of the Web application, are less interested in information at a detailed level of granularity, such as the error classification, and are more interested in summary information and general measures of accessibility. Instead, those who access the validation tools more frequently are typically more involved in actually modifying their Web applications, and therefore need more detailed information. Likewise, from the survey, it emerged that those who use the tools more frequently find it more useful that the tools provide indications on their limitations, compared to those who use them less frequently. This could be explained by the fact that frequent users find it more useful to have an explicit indication of their limitations (an aspect that tools tend not to emphasize much) to better understand their actual abilities in performing the validation, and thereby to more easily identify the areas to which further attention should be posed. Also the test indicated that users often appreciate the possibilities to receive reports targeted to the various types of possible uses, and simply providing basic data may be insufficient, as it is often necessary to consider the actual audience. Thus, information should be provided in a way to be correctly interpreted and understood by the audience and able to consider the different stakeholders' needs. For example, people who have to modify the implementation appreciate the possibility to receive clear indications associating errors and code lines that generate them, and how to correct them. Moreover, reports should be interactive, with the possibility of filtering the results in order to see only the information that the user judges as most significant, and providing a preview of the target page together with the errors. In some tools (e.g. Lighthouse) the report was considered too cursory. Regarding QualWeb, some user pointed out that the graphical summary of results was insufficient; thus, more graphical representations (e.g. pie charts, bar charts, etc.) summarising validations results would be appreciated.

Current coverage of automatic validators and better awareness of their role. One aspect that emerged is that there is a need for the right expectations in validation tools. On the one hand, among the aspects that users would value most in terms of transparency is that they would like to have more specific and precise information on the actual tests that the tools currently perform, to have a better overview of *what has been (or has not been) checked in the current version of the tool*. Thus, the current coverage must be precisely indicated because the output provided by the automatic validators often represents the starting point of manual checks, thereby users need to have full awareness of what still needs to be done by them. In addition, since often users exploit several tools (which could provide different results on the same page), this point seems particularly relevant especially in case of non-expert users, who sometimes might not have the knowledge and the skills to understand the causes of such inconsistencies on their own.

On the other hand, many users highlighted the importance of further emphasising that accessibility tools should clearly state upfront that there are some aspects that they cannot assess, clearly indicating what they are, and that manual checks should always be included when evaluating the accessibility of a page. In this regard, some users suggested further improving the tools so that they can provide users with concrete indications about how to carry out such manual tests.

Need to provide details about how an accessibility check is implemented. Indicating what success criteria or techniques are supported may not be sufficient. Users also need to know how they are supported, since sometimes tools interpret some of them differently, which can generate false positives and false negatives. In order to increase the confidence in their behaviour, the tools should clearly indicate for each technique what elements they analyse and how. By analysing the users' feedback in the survey, it comes to light that also developers and accessibility experts have some difficulties interpreting the reasons behind errors returned by a tool. One motivation for this problem is that even after carefully reading the W3C documentation about a violated technique, some issues remain in understanding how such technique should be implemented in the validator. The WCAG technique definitions are defined with descriptions, which sometimes tool developers can interpret differently. For this reason, in order to be more transparent, a tool should also expose to end-users how it implemented a validation technique; in this way, it could be easier to provide users with a solution for the accessibility issue or identify issues in tools' implementation. To this regard, some users mentioned that ACT rules can help, as a uniform format for writing accessibility test rules in order to better document and share the used testing procedures in a non-ambiguous manner. However, ACT rules still provide only partial coverage of the possible accessibility tests, and in any case the documentation about how tests are performed should be provided in a way understandable also by non-technical people.

Provide information about how to solve accessibility problems. This is another crucial point recommended by users. Reporting only the list of accessibility issues with the correspondent technique (or success criterion) description and the link to the W3C documentation may not be sufficient. Developers need clear and practical indications on how to modify the code to be compliant with the technique, with examples relevant to the issue at hand. Some tools provide some support in this direction but the examples shown are fixed, and sometimes are not particularly useful in solving the current problem.

Better connecting guidelines validation reports with actual user experience, in particular of disabled users. The need for properly connecting the results of the validation tools with the actual experience of users of the considered Web application, in particular those who are disabled, emerged as an important aspect. Thus, it would be useful if tools provided support in relating the validation results to the concrete problems that (a specific subset of) disabled users can have when a particular criterion is violated, which could be especially beneficial for users that do not have sufficient knowledge about the impact that an accessibility violation could have on specific disabilities. Another possible approach [Salehnamadi et al., 2021] to consider the user experience is to focus the validation on a subset of the Web content, which is considered most important for the user, since it is more closely related to the most frequent tasks carried out with the considered Web application, and thereby avoiding producing a massive amount of accessibility warnings that can disorient end users.

Better awareness of tools' updates. Accessibility validators inevitably need to dynamically evolve over time. One of the main reasons is that they should be able to follow the evolution of the technologies that can be used to implement Web pages (e.g. a tool can support a new technology, or even a new version of the same technology). In order to be transparent, tools need to provide users with precise overview of *which* significant changes in their implementation have been made and also *when* they were made, also to heighten users' confidence about whether the tool is maintained in a sufficiently up-to-date manner.

Support effective communication with its users and enhance user's participation in the development of the tools. Sometimes tools are perceived as non-transparent not only to outsiders, but also to experts. Thus, effective communication support with users would be helpful to explain how the tool works in some cases, and also as a feedback mechanism through which users can express some concerns they have about the functioning of the tool. In this regard, providing users with the possibility and the means to more actively participate and intervene in the development work could be beneficial to boost the level of adoption of the tool itself.

Need to increase the knowledge of accessibility within organisations. One further reflection concerns that the validation of accessibility guidelines is becoming more and more complex. The WCAG 2.1 have 78 success criteria and more techniques associated with them. The validation of each of them requires specific algorithms to analyse the relevant elements in Web applications. Understanding all such aspects requires some effort and time. Unfortunately, often organizations aim to declare that they support accessibility without actually dedicating sufficient human resources to it, and the people involved in its validation can dedicate little time to this activity, thus leading to further difficulties in understanding the various relevant aspects.

Limitations of the study. Our work is not without shortcomings and limitations, which however can show the potential for future work. One limitation is that among the users involved in the survey and test there was no complete balance in the knowledge of the tools considered. In addition, in the second study, in order to avoid a too long test, we decided to limit the analysis to only three tools.

8 CONCLUSIONS AND FUTURE WORK

In this paper we present some design criteria for supporting transparency in automated Web accessibility tools, and how such criteria are currently supported by a set of existing tools. In order to test the relevance of such criteria we also carried out a survey and a user test, which have provided useful feedback confirming that if such criteria were fully applied, they would improve the user experience when using such accessibility validation tools, and their trust in them. Indeed, in the survey the users ranked positively the various aspects proposed by the design criteria, and it also emerged that the type of information they need and appreciate depends on their actual use of such tools. The user test confirmed that current tools do not fully support such criteria since users sometimes complained about missing or unclear information, and gave responses that demonstrated that they did not completely understand the provided results. Indeed, one aspect that emerged is that transparency should not consist in providing long lists of data about how the tool works in all its details (which sometimes can generate confusion rather than providing clarifications), rather, the information provided by tools should be easy to understand and, especially, match the needs of different stakeholders (who use the tools differently, and have various expectations, knowledge and perception) and be practically relevant to them in the various situations in which they will require support from accessibility validators.

The approach to accessibility validators tools has similarities with those adopted in the artificial intelligence area, since also in this case the aspects to clarify are the input and output of the systems, how their results are obtained, and what users can do to change them. The difference consists in the actual information they use, how they process it, and present the results to the relevant stakeholders.

Thus, we structured the design criteria proposed according to a number of aspects ranging from the standards, success criteria and techniques supported, how the tool categorizes the errors found, how the reported information is organized, to whether the tool is able to assess the accessibility in specific cases (e.g. dynamic pages). Such aspects can represent and be used as a logical framework for tool developers and users to characterize the tools in terms of transparency, and also be used by developers and practitioners to better integrate transparency-related considerations in their work, to avoid pitfalls when developing and deploying accessibility validators. Indeed, since we found that such aspects are not fully supported by existing tools, they can also be helpful in drafting future work for tool developers to improve the transparency of their accessibility evaluation tools ‘by design’.

REFERENCES

J. Abascal, M. Arrue & X. Valencia. (2019). Tools for Web accessibility evaluation. Web Accessibility (pp. 479-503). London: Springer.

Accessibility Scanning & Monitoring, by UserWay, https://userway.org/scanner (Last accessed 30 July 2021)

Accessi.org by Adam, https://www.accessi.org/ (Last accessed 30 July 2021)

aCe by accessiBe, https://accessibe.com/ (Last accessed 30 July 2021)

A11y Color Contrast Accessibility Validator , https://color.a11y.com/?wc3 (Last accessed 1 March 2022)

S. G. Abduganiev. 2017. Towards automated Web accessibility evaluation: a comparative study. Int. J. Inf. Technol. Comput. Sci.(IJITCS) 9, 9 (2017), 18–44.

S. Abou-Zahra. 2017. Evaluation and Report Language (EARL). Retrieved February 2, 2017 from https://www.w3.org/TR/EARL10-Schema/#OutcomeValue

ACT Rules, https://act-rules.github.io/pages/about/ (Last accessed 30 July 2021)

M. Arrue, M. Vigo & J. Abascal. 2008. Including heterogeneous Web accessibility guidelines in the development process. IFIP International Conference on Engineering for Human-Computer Interaction (pp. 620-637). Berlin: Springer.

M. Ballantyne, A. Jha, A. Jacobsen, J.S. Hawker, & Y.N. El-Glaly. 2018. Study of Accessibility Guidelines of Mobile Applications. 17th International Conference on Mobile and Ubiquitous Multimedia (pp. 305-315). ACM.

A. Beirekdar, J. Vanderdonckt, & M. Noirhomme-Fraiture. 2002. Kwaresmi–Knowledge-based Web Automated Evaluation with REconfigurable guidelineS optiMIzation. (Springer, Ed.) DSV-IS, 2545, 362-376.

A. Beirekdar, M. Keita, M. Noirhomme, F. Randolet, J. Vanderdonckt J, C. Mariage. 2005. Flexible Reporting for Automated Usability and Accessibility Evaluation of Web Sites. In: Costabile M.F., Paternò F. (eds) Human-Computer Interaction - INTERACT 2005. INTERACT 2005. Lecture Notes in Computer Science, vol 3585. Springer, Berlin, Heidelberg

G. Brajnik. 2004. Comparing accessibility evaluation tools: a method for tool effectiveness. Universal access in the information society 3, 3-4 (2004), 252–263.

G. Brajnik, Y. Yesilada, & S. Harper. 2012. Is accessibility conformance an elusive property? A study of validity and reliability of WCAG 2.0. ACM Transactions on Accessible Computing (TACCESS), 4(2), 1-28.

G. Brajnik & M. Vigo. 2019. Automatic Web Accessibility Metrics. Where we were and where we went. (Springer, Ed.) Web Accessibility, 505-521.

A. Burkard, G. Zimmermann, and B. Schwarzer. 2021. Monitoring Systems for Checking Websites on Accessibility. Frontiers in Computer Science 3 (2021), 2.

EqualWeb, https://www.equalweb.com/ (Last accessed 30 July 2021)

EU Commission. (2016, October 26). Directive (EU) 2016/2102 of the European Parliament and of the Council. Retrieved from https://eur-lex.europa.eu: https://eur-lex.europa.eu/eli/dir/2016/2102/oj

Accessibility Checker by EXPERTE, https://www.experte.com/accessibility, (Last accessed 30 July 2021)

N. Fernandes, N. Kaklanis, K. Votis, D. Tzovaras, & L. Carriço. 2014. An analysis of personalized Web accessibility. Proceedings of the 11th Web for All Conference (p. 19). ACM.

T. Frazao and C. Duarte. 2020. Comparing accessibility evaluation plug-ins. In Proceedings of the 17th International Web for All Conference (W4A '20). Association for Computing Machinery, New York, NY, USA, Article 20, 1–11. DOI:https://doi.org/10.1145/3371300.3383346

Free Web Accessibility Checker, AlumniOnline Web Services, https://www.alumnionlineservices.com/scanner/, (Last accessed 30 July 2021)

J.L. Fuertes, R. González, E. Gutiérrez & L. Martínez. 2009. Hera-FFX: a Firefox add-on for semi-automatic Web accessibility evaluation. Proceedings of the 2009 International Cross-Disciplinary Conference on Web Accessibility (W4A) (pp. 26-35). ACM.

G. Broccia, M. Manca, F. Paternò, and F. Pulina. 2020. Flexible automatic support for Web accessibility validation. Proceedings of the ACM on Human-Computer Interaction 4, EICS (2020), 1–24.

G. Gay and C. Qi Li. 2010. AChecker: open, interactive, customizable, Web accessibility checking. In Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A). 1–2.

J. Gulliksen, H. Von Axelson, H. Persson, & H. Göransson. 2010. Accessibility and public policy in Sweden. Interactions, 17(3), 26-29.

M. Hellmann, D. C. Hernandez-Bocanegra and J. Ziegler, Development of an Instrument for Measuring Users', Perception of Transparency in Recommender Systems, Joint Proceedings of the ACM IUI Workshops 2022, March 2022, Helsinki, Finland

IBM Equal Access Accessibility Checker, https://www.ibm.com/able/toolkit/tools/, (Last accessed 30 July 2021)

M. Y. Ivory, J. Mankoff, and A. Le. 2003. Using automated tools to improve Web site usage by users with diverse abilities. Human-Computer Interaction Institute (2003), 117.

L. R. Kasday. 2000. A tool to evaluate universal Web accessibility. In Proceedings on the 2000 conference on Universal Usability. 161–162.

J. Lazar & A. Olalere. 2011. Investigation of best practices for maintaining section 508 Compliance in US federal Web sites. International Conference on Universal Access in Human-Computer Interaction (pp. 498-506). Berlin: Springer.

Q. V. Liao, D. M Gruen, S. Miller, Questioning the AI: Informing Design Practices for Explainable AI User Experiences, CHI 2020

MAUVE++, https://mauve.isti.cnr.it/ , (Last accessed 30 July 2021)

A. Miniukovich, M. Scaltritti, S. Sulpizio, and A. De Angeli. 2019. Guideline-Based Evaluation of Web Readability. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19). Association for Computing Machinery, New York, NY, USA, Paper 508, 1–12. DOI:https://doi.org/10.1145/3290605.3300738

S. Mirri, L.A. Muratori, & P. Salomoni. 2011. Monitoring accessibility: large scale evaluations at a geo political level. The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility (pp. 163-170). New York: ACM.

A. M. Molinero, F. G. Kohun, and R. Morris. 2006. Reliability in Automated Evaluation Tools for Web Accessibility Standards Compliance. issues in Information Systems 7, 2 (2006), 218–222.

L. Moreno, R. Alarcon, I. Segura-Bedmar, and P. Martínez. 2019. Lexical simplification approach to support the accessibility guidelines. In Proceedings of the XX International Conference on Human Computer Interaction (Interaccion '19). Association for Computing Machinery, New York, NY, USA, Article 14, 1–4. DOI:https://doi.org/10.1145/3335595.3335651

J. Mucha, M. Snaprud, & A. Nietzio. 2016. Web page clustering for more efficient website accessibility evaluations. International Conference on Computers Helping People with Special Needs (pp. 259-266). Springer.

A. Nietzio, M. Eibegger, M. Goodwin, & M. Snaprud. 2011. Towards a score function for WCAG 2.0 benchmarking. Proceedings of W3C Online Symposium on Website Accessibility Metrics. Retrieved from https://www.w3.org/WAI/RD/2011/metrics/paper11

M. Padure and C. Pribeanu. 2019. Exploring the differences between five accessibility evaluation tools. (2019).

P. Parvin, V. Palumbo, M. Manca, F. Paternò. 2021. The Transparency of Automatic Accessibility Evaluation Tools. In Proceedings of the 18th International Web for All Conference (W4A '21), April 19–20, 2021, Ljubljana, Slovenia. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3430263.3452436

F. Paternò, A. Schiavone, The role of tool support in public policies and accessibility. ACM Interactions 22(3): 60-63 (2015)

J. Pelzetter, A Declarative Model for Web Accessibility Requirements and its Implementation. Frontiers Comput. Sci. 3: 605772 (2021)

H. Petrie, N. King, C. Velasco, H. Gappa, G. Nordbrock. The usability of accessibility evaluation tools. In: Stephanidis, C. (ed.) UAHCI 2007. LNCS, vol. 4556, pp. 124–132. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-73283-9_15

C. Power, A. Freire, H. Petrie & D. Swallow. 2012. Guidelines are only half of the story: accessibility problems encountered by blind users on the Web. Proceedings of the SIGCHI conference on human factors in computing systems (pp. 433-442). ACM.

Puppeteer, https://github.com/puppeteer/puppeteer (Last accessed 30 July 2021)

QualWeb, http://qualweb.di.fc.ul.pt/ (Last accessed 30 July 2021)

A. Schiavone, F. Paternò, An extensible environment for guideline-based accessibility evaluation of dynamic Web applications, Universal Access in the Information Society, Springer Verlag, 14(1): 111-132, 2015.

N. Salehnamadi, A. Alshayban, J.-W. Lin, I. Ahmed, S. Branham, and S. Malek. 2021. Latte: Use-Case and Assistive-Service Driven Automated Accessibility Testing Framework for Android. In CHI Conference on Human Factors in Computing Systems (CHI '21), May 8– 13, 2021, Yokohama, Japan. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3411764.3445455

TAW, by CTIC Technology Centre, https://www.tawdis.net/ (Last accessed 30 July 2021)

M. Vigo, J. Brown, and V. Conway. 2013. Benchmarking Web accessibility evaluation tools: measuring the harm of sole reliance on automated tests. In Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility. 1–10.

Y. Yesilada, G. Brajnik, M. Vigo, S. Harper: Exploring perceptions of Web accessibility: a survey approach. Behav. Inf. Technol. 34(2), 119–134 (2015)

WAVE, https://wave.webaim.org/ (Last accessed 30 July 2021)

W3C WAETL, Web Accessibility Evaluation Tools List, https://www.w3.org/WAI/ER/tools/ (last accessed 20 October 2021).

APPENDIX – SURVEY QUESTIONS

Indicate your gender

Indicate your country

Indicate your age

In which sector do you work?

Approximately, how many employees does your organization include

Which role most identifies you?

Do you use automated accessibility assessment tools to support your work?

- Name of the first tool
  - How often did you use it?
- Name of the second tool
  - How often did you use it?
- Name of the third tool
  - How often did you use it?

How would you define the transparency of automatic accessibility assessment tools?

- That the tool states what standards, success criteria and techniques it supports in the assessment
- That the tool specifies how it categorizes evaluation results (errors, warnings, etc.)
- That the tool is able to provide general measures that make explicit the level of accessibility of the website/mobile app
- That the tool presents the evaluation results both in a more summarized way (e.g., graphs, tables, etc.) and in a more detailed way (e.g., code view)
- That the tool gives some practical indications on how to resolve the detected problem
- That the tool gives some indication of its limitations
  - For example, which limitations?

Have you ever been unable to understand the results of an accessibility evaluation performed by an automated tool?

- If YES, do you remember what kind of difficulties you encountered?

Are there any other features you think an automated accessibility evaluation tool should have in order to be transparent?

- If YES, which ones?