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, including 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 tools. We also report on some empirical feedback gathered through a survey and a user test on the users' perception of the transparency of current tools, which support the adoption of such design criteria.

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 [Gullisken 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 in 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], automatic tools still play a crucial role in ensuring the accessibility of websites. They help human operators collect and analyse data about the actual application of accessibility guidelines, detect non-compliance, and provide relevant information about addressing the possible problems. Over time many tools have been put forward in this area (e.g. [Beirekdar et al. 2012], [Bereikdar et al., 2015]). As of October 2021, the W3C Web Accessibility Evaluation Tools list[1] contains 159 elements. However, for various reasons most of them had limited impact. 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, [Miniukovic et al. 2019] has focused on readability issues, and provides some automatic support but only for assessing some text-related properties, or the A11y Color Contrast checker[2] mainly focuses on checking the contrast in a Web page. Another example is the contribution [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 other tool developers, we often noticed that such tools 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 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, the unclear expression of these differences makes them be perceived in different ways by users, sometimes they are misinterpreted, and can generate misunderstandings and lack of trust in automatic validation tools.

Better 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

[1] https://www.w3.org/WAI/ER/tools/
[2] https://color.a11y.com/?wc3

some criteria that can be used to analyse it, and provided an initial analysis of four validation tools according to them.

By transparency of an accessibility validation tool, we mean its ability to clearly indicate to its users what accessibility aspects it is able to validate, the meaning of the results generated, and its limitations.

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 the need to continuously update their support for the validation of the associated accessibility guidelines. Moreover, the W3C WCAG accessibility guidelines are defined in a high-level format that should be interpreted by tools 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 to 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 in recent times. 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, and 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.

In this paper, we define the transparency concept and indicate some design criteria to support it, provide an analysis of a set of accessibility validation tools according to such criteria, investigate users' perception of tools transparency through a survey and a user test, and finally conclude with some discussion and recommendations. To analyse how a set of validation tools support the transparency criteria, we have 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 and freely accessible on the Web. The empirical feedback on the transparency issues has been obtained through a survey and a user test. The survey collects the opinion of 138 people who have used such tools and are actively involved in the accessibility topic. 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 on the code of Web sites). The user test involves eighteen people who have been asked to perform a typical set of tasks in accessibility validation using three different tools, then they have 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.

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 evaluations 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, 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 previous 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 apps 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 from evaluation engine to evaluation engine. 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 the 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.

3 ASPECTS RELEVANT FOR TRANSPARENCY OF ACCESSIBILITY VALIDATION TOOLS

The criteria we propose derived from a previous preliminary analysis [Parvin et al., 2021], 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.

As stated, transparent tools should enable users to make fully informed decisions based on a clear understanding of how the automatic validation tools work. In general, in order to be transparent, an accessibility validation tool should be clear about what it is able to check (and consequently what it is not able to check), provide its results at different levels of granularity (from the checking of the basic elements of a Web page to overall measures of the accessibility of the considered pages), also considering that they may be seen by people with different roles and expertise, and should indicate helpful information to resolve the detected problems.
For such reasons, in order to be transparent, an automated validation tool should make explicit the following information on its operations:

- *Which 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 82 success criteria and many associated techniques (with this set that dynamically changes), and some of them cannot be automatically validated at all. Thus, users need to understand the current coverage of the considered tools.
- *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 the user. 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.
- *How the reported information is provided by the tool.* For this purpose, it is important whether such information is provided with varying granularity levels, and using different types of presentations.
    - 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.
    - 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.
- *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 in fixing the identified accessibility violations. Clearly, useful additions would be to provide 'repairing' functionalities that assist users through the process of fixing some accessibility problems, by providing suitable recommendations for solutions.
- *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 validations 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. 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

In order to perform the analysis of the transparency of accessibility tools, we selected a sub-set 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.For example, 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 licence 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 has released such guidelines 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, which is the informal language of the Internet.
- **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 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. In the next sections (4.1-4.11) we will introduce each of them, indicating how it addresses the aspects relevant from a transparency perspective. Then, in Section 4.12 we summarise their main characteristics, also considering the transparency criteria presented above in Table 1.

4.1 Ace Accessibility Checker by accessible[3]

The Ace Accessibility Checker 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.

Users can enter the URL of the Web page that they want to evaluate, and 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. Then, a numeric score is provided for each considered category (Clickable, Titles, Orientation, Menus, Graphics, Forms, Document, Readability, Carousel, Tables, General).

A live preview of the evaluated Web page is provided; however, there are no references to the errors discovered by the tool (see Figure 1).

For each error, the tool provides the number of analysed elements, and the number of success and failures along with the corresponding requirement. However, such elements are not related to the correspondent WCAG technique. For each accessibility issue, it shows a code snapshot of some failed and successful elements (the code snapshot section is not complete because it shows only a subset of all the detected issues). It is also possible to receive the accessibility report by mail; the report contains the list of the checkpoints considered during the evaluation, the number of elements with result success and failure, and the code snapshots of some failed elements. The report does not provide any preview of the evaluated UI and no link between the checkpoints analysed during the evaluation and the corresponding WCAG technique.

[3] https://accessibe.com/

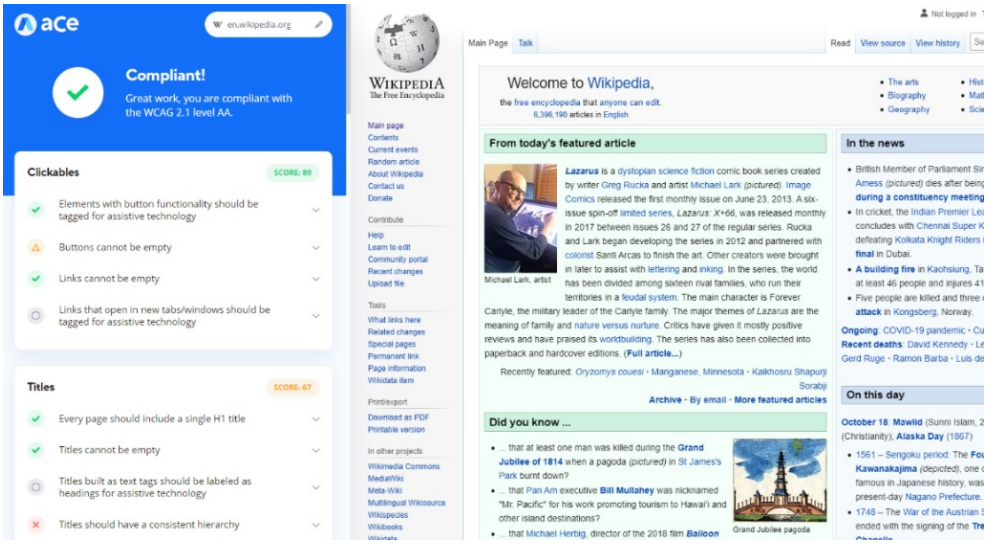

Figure 1:The ACE Tool by AccessiBe

4.2 Accessi.org[4]

The Accessi.org tool supports validation of WCAG 2.0 and 2.1 of single pages. Differently from other analysed tools, 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. The priority can assume the following values: "high impact errors" which means your Web users will find those pages "impossible" to use, hence, will require manual review. The medium-impact errors make the pages "difficult" for people with disabilities while the low-impact errors are "somewhat difficult", containing errors such as improper link text and lack of skip to content link. It is also possible to obtain a report as a webpage and pdf format, and to generate a simple accessibility statement. The result report is structured in terms of success criteria violated. For each success criteria there is a link showing the elements that generated the error, a text explaining the issue and a link to the W3C related technique. 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 (Figure 2). No hints are provided regarding the support of dynamic pages.

[4] https://www.accessi.org/

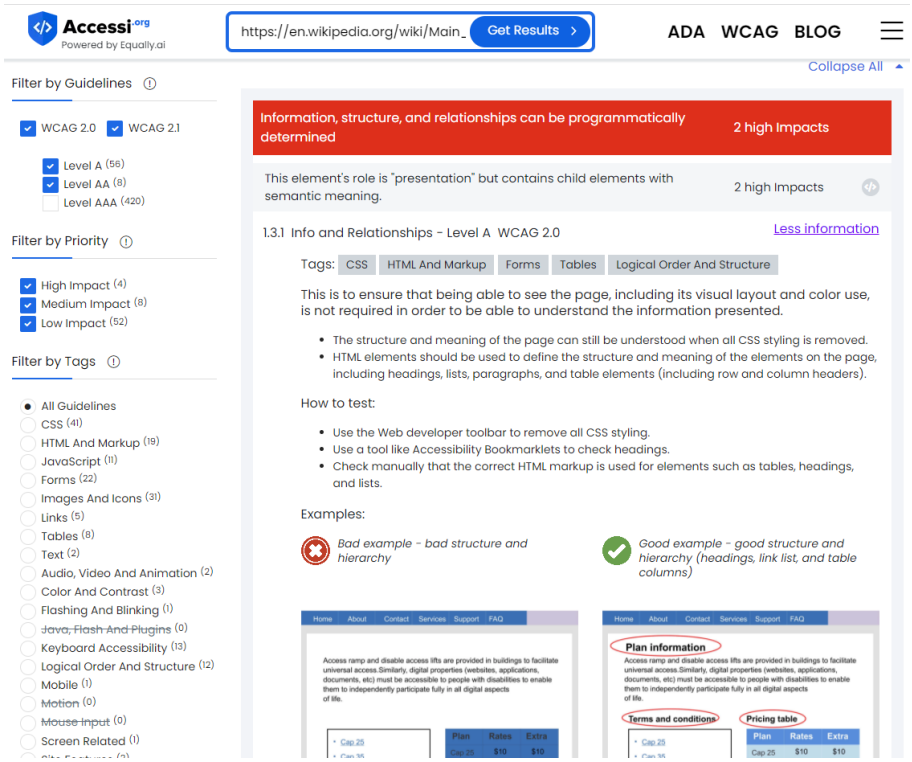

Figure 2: The Accessi Tool

4.3 Accessibility Scanner by UserWay[5]

The UserWay Accessibility scanner tool has two versions: The Free scan supports only single page validation and analysis for desktop interfaces; and the Pro version supports Web sites validation also for mobile interfaces. The FAQ section introduces the WCAG structure and states that the validator supports the version 2.1; however, to understand how UserWay Accessibility Scanner supports the success criteria and techniques, it suggests to contact the customer care team, while from a transparency viewpoint a tool should immediately expose which criteria and techniques it supports. The validation report shows the accessibility violations grouped by level of conformance of the WCAG 2.1 through a dashboard that indicates the number of tests executed, and how many of them are passed, failed, and not applicable (Figure 3). There is also a classification in terms of severity, but it is unclear how it has been defined.

[5] https://userway.org/scanner

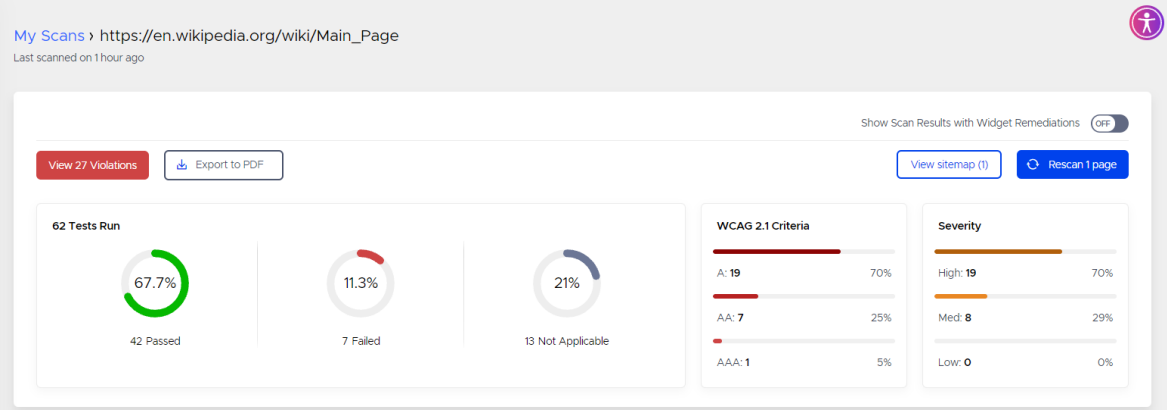

Figure 3:The UserWay Tool

4.4 Equalweb[6]

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. Validation results are classified into 'issue remains', 'issues passed' and 'review only'. The meaning of the classification can be understood, but no specific indication of its meaning is given.

Two general measures are offered: the Overall accessibility Score, which should indicate the percentage of accessible content on the page, and Assessment Statistics, calculated on all the pages scanned on the site (Figure 4). The results are displayed as an expandable list of elements, and for each of them a description of the technique is provided, along with a "context and selector" section containing the part of the HTML code that addresses the problem, and the xPath to that code. The tool offers the possibility to share the report via link, WhatsApp, email or to download it in pdf format. In the technique info section, guidance is given on how to address the relevant technique, with a link to the relevant page on the W3C site. A list of validation results per scanned page is provided in the case of multipage validation. For each scanned page, the number of “fixed errors”, “remaining errors”, “contrast errors”, “ARIA attributes”, “Role attributes” are shown. There is no detailed documentation about the dynamic Web page evaluation support, however, we can suppose that such pages are supported through the plugin which supports the evaluation of the Web page currently loaded on the browser.

[6] https://www.equalweb.com/

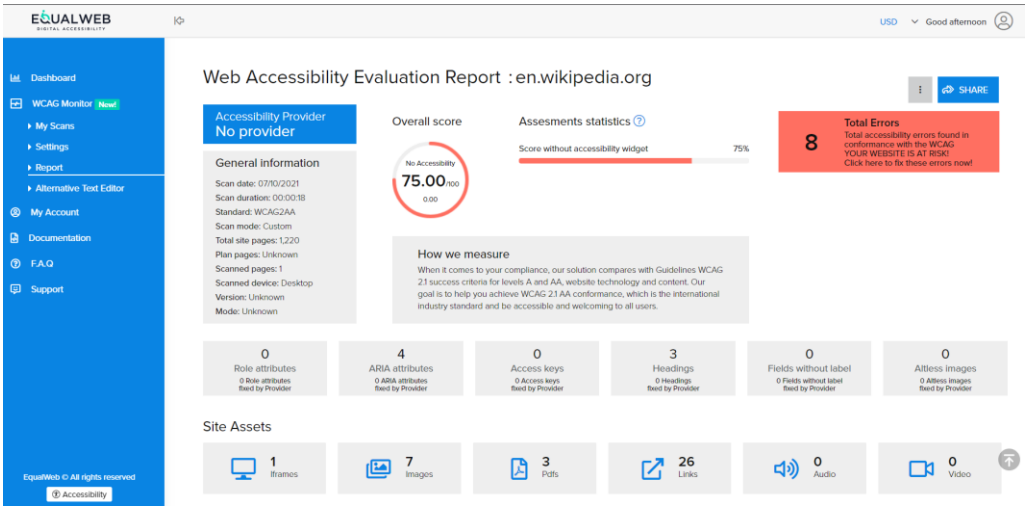

Figure 4:The EQUALWEB Tool

4.5 EXPERTE[7]

The EXPERTE's Accessibility checker supports the multipage evaluation through a crawler able to discover all the pages composing a website; the discovered Web pages will be then evaluated against 41 features across 8 categories (Navigation, Aria, Name and Labels, Contrast, Tables and List, Audio and Video, Internationalization and Localization). The tool lists all the accessibility checkpoints examined during the validation (it also shows the 'passed' and 'not applicable' checkpoints); however, it does not indicate the corresponding W3C WCAG technique or success criteria.

The accessibility issue descriptions are closely related to WCAG techniques, but the provided documentation is not linked to the W3C website, since it refers the WebDev.com website; there are no references to the code line, and no example is provided to help the users in better understanding the accessibility error and then solve the issue. Users cannot provide any preference about the target device and viewport dimensions. The evaluation is based on Lighthouse, an open-source tool, which provides a score (from 0-100) indicating the site's technical accessibility (Figure 5).

Compared to the other presented tools, we can consider EXPERTE as a tool that addresses some transparency aspects, however the features considered during the validation process should be more closely related to the W3C standard guidelines (WCAG 2.1).

[7] https://www.experte.com/accessibility

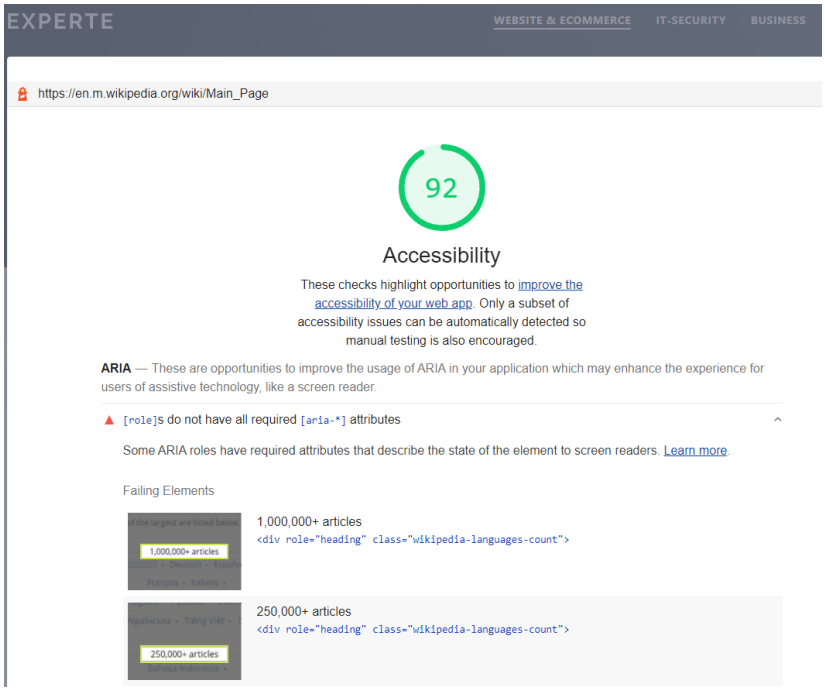

Figure 5:The Experte Tool

4.6 Free Web Accessibility Checker by AlumniOnline[8]

The official site of Free Web Accessibility Checker by AlumniOnline Web Services mainly presents a plugin to address accessibility problems on WordPress websites; however, it also offers a scan for single pages (it is not possible to evaluate an entire website). The Features section describes the plugin, and it states that 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. However, the title of the scan page is "Free WCAG 2.0 and Section 508 Web Accessibility Check", thus it seems not clear whether the page is validated against WCAG 2.0 or 2.1. This scan generates a report of results (Figure 6) containing: a list of errors with a description of the problem and the corresponding code, an issue summary showing the number of elements of a detected problem, and finally, a link to the W3C website explaining the corresponding success criteria. It is possible to view and then download the generated report in pdf format. No preview and no metrics are provided.

[8] https://www.alumnionlineservices.com/scanner/

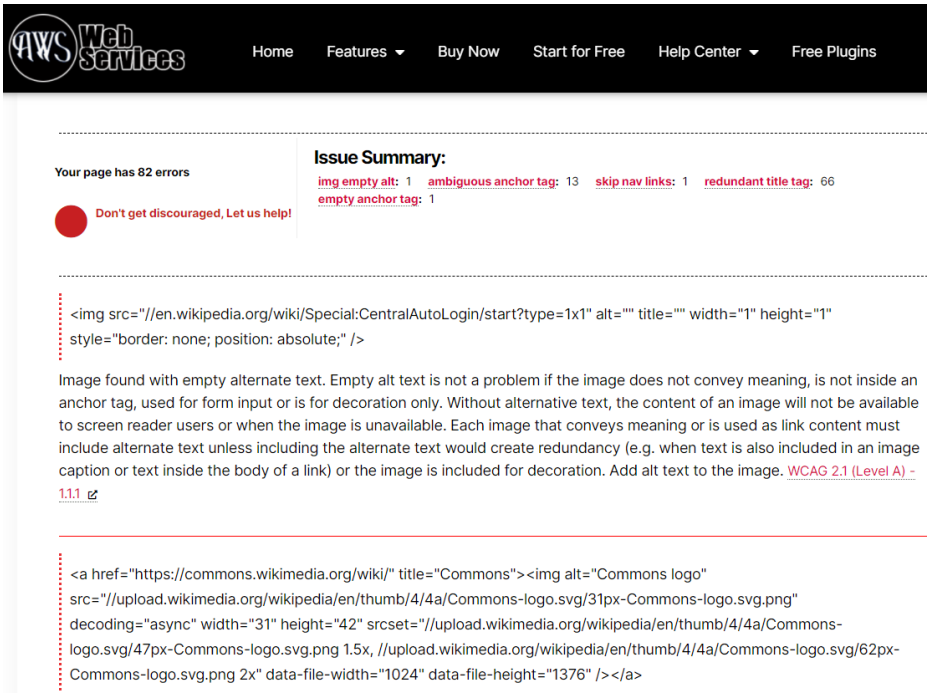

Figure 6:The Free Web Accessibility Checker by AlumniOnline Tool

4.7 IBM Access Accessibility Checker[9]

IBM 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. By accessing the extension's "Options", it is possible to select the accessibility guidelines to consider among IBM Accessibility, WCAG 2.1 (A, AA) and WCAG 2.0 (A,AA).

The tool categorizes accessibility issues into: i) Violations: accessibility failures that need to be corrected; ii) Needs review: issues that may not be a violation, so a manual review is needed; iii) Recommendations: opportunities to apply best practices to improve accessibility further. For each category of issues, the tool provides the number of issues found in the validation result.

It provides summary information about the accessibility of the tested page in terms of the percentage of elements with no detected violations or items to review (Figure 7). By selecting each specific issue, it is possible to get further information about the associated technique(s), the concerned element in the code (with the possibility to copy the associated code line), the recommended remediation (i.e., what to do to fix the error), and which category of users it affects.

[9] https://www.ibm.com/able/toolkit/tools

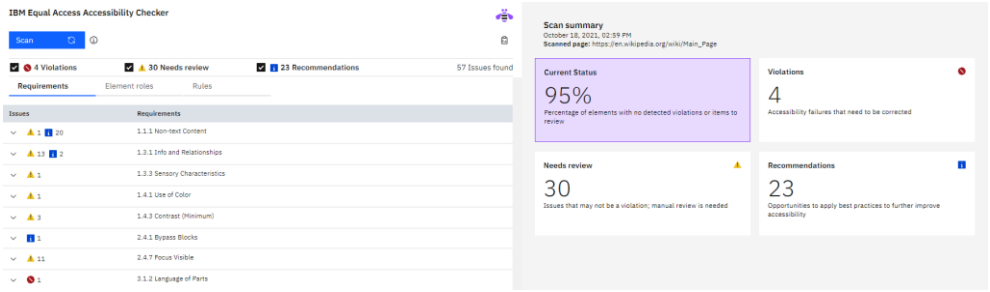

Figure 7: The IBM Access Accessibility Checker Tool

4.8 MAUVE++[10]

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). In the *Info* section, it exposes to the users the list of supported techniques. In the same section, there is an explanation of the different possibilities that the tool offers regarding the validation of Static Web Pages vs Server-Side Rendering Validation: in the traditional Static Web Page validation, the validation engine downloads the HTML and the CSS of the page, then it parses and validate the corresponding DOM.

Using the Server-Side Rendering Validation, the validation engine does not parse the static Web page code; indeed, it exploits the Puppeteer library to load the HTML (and CSS) code within a headless version of the Chrome browser. In this way, it simulates the loading phase as if the page were open in the user's browser. This can be useful in the case of Single Page Applications or in high dynamic Web pages populated through external services calls.

In order to indicate the overall level of accessibility detected MAUVE++ provides two metrics (Figure 8): 1) Accessibility Percentage which indicates how much the website is accessible in terms of the number of checkpoint successfully evaluated over the total number of evaluated checkpoints for which the tool has been able to make a decision (fail or pass); 2) Evaluation Completeness, indicating the percentage of evaluated checkpoints for which the tool has been able to make a decision (fail or pass) about the validation.

Moreover, MAUVE ++ generates the evaluation reports in Web, PDF and Earl formats. Regarding the Web report, it is provided with two different views: Web developer view (code-oriented style) and End-user view (for people without programming experience), which shows errors and warnings through charts and tables.

The validation report categorizes the issues in Error, representing an accessibility violation that can be detected automatically; Warning, representing a possible problem that cannot be verified automatically and needs a manual review. Finally, the Success category is associated with the elements that passed the test. MAUVE++ can also evaluate the accessibility of entire websites. It also provides a preview of the Web page with the possibility to highlight the points where the accessibility problems occurred.

[10] https://mauve.isti.cnr.it/

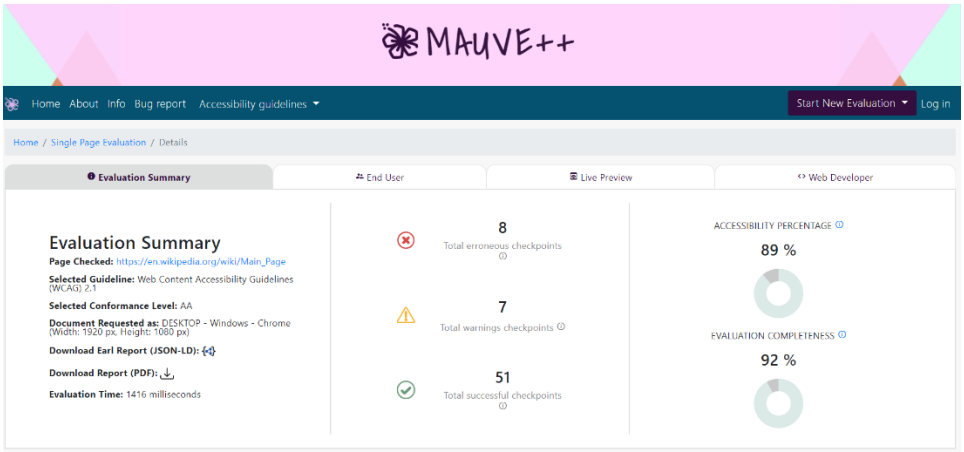

Figure 8:The MAUVE++ Tool

4.9 QualWeb [11]

QualWeb is an open-source automated Web accessibility evaluation service that incorporates contributions from different research projects and efforts. A user can automatically check a Web page directly by inserting its URL, but it does not provide any opportunity to evaluate an entire website. 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). Although this information is accessible for technical users in the GitHub repository at the *https*: *//github.com/qualweb*, the tool gives limited information to non-technical users regarding the guidelines supported in the tool's "About" section. Regarding the support for dynamic page evaluation, they do not provide any information; however, by analysing the report provided for a single Web page, it seems the tool can evaluate it correctly through a server-side rendering (SSR) capability.

QualWeb provides the evaluation report only through the website, and it is not possible to download it. The report consists of several sections (Figure 9): The *summary* shows the total number of errors. It also allows users to filter the results that match their particular needs. The *Evaluation Report* gives a complete report of all the errors. The tool's report includes the description and the results of the tested rules (i.e., passed, failed, warning and not applicable), a link to the full description of the rule at the *https*: *//www.w3.org/* website, the related success criteria along with the priority level, and finally, the HTML code line related to the recognized issue. A failure occurs when the tool can detect automatically and unambiguously if a given HTML element has an accessibility problem. A pass, is generated from elements that, unambiguously, are classified as having no accessibility problems. A warning issue ensues when the tool can partially detect accessibility problems, requiring additional inspection (often by experts). A Not Applicable issue occurs when there are no relevant elements on the Web page to be tested. Another relevant functionality in the report section of this tool is the Visual Representation button, which triggers the preview of the elements in the Web page that generated the error; however, it shows the preview of the single element without any context, which makes it difficult to interpret the preview.

While QualWeb clearly shows the evaluator's settings, it does not provide the possibility of selecting the conformance level.

[11] http://qualweb.di.fc.ul.pt/

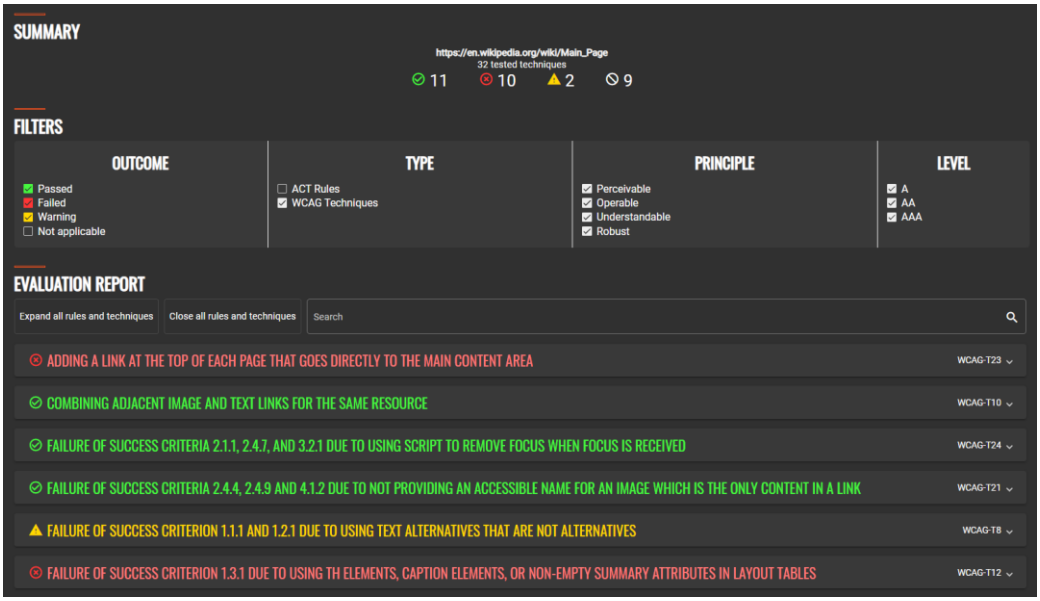

Figure 9:The QUALWEB Tool

4.10TAW[12]

TAW is a free automatic online tool for analyzing 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 have not found 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. There is also a desktop version, but it supports only WCAG 1.0. It supports single-page validation, validating the accessibility of HTML, CSS and also the code produced by JavaScript. The TAW checks in the analysis fall into two categories: i) Automatic: problems of accessibility that the tool detects by itself and that they must be solved; ii) Manual: the tool indicates the existence of a possible problem that the evaluator must confirm or discard. It is possible to receive the detailed result of the analysis in the email sent to the user (it is a Web page, thereby accessible also via the browser if it is not possible to see it correctly in the email). The result of the validation of a page is a summary (Figure 10) that (in its top part), categorises the issues into three groups: i) Problems (=corrections are needed); Warnings (= a human review is necessary); Not reviewed (=a fully manual review is needed). It is specified for each type of issue: how many occurrences were found, in how many success criteria, and in how many principles. In addition, it is possible to get a detailed report via email where it shows a table with all the criteria grouped by principle, indicating if you need manual validation or if not, the number of errors or warnings they have given.

[12] https://www.tawdis.net/

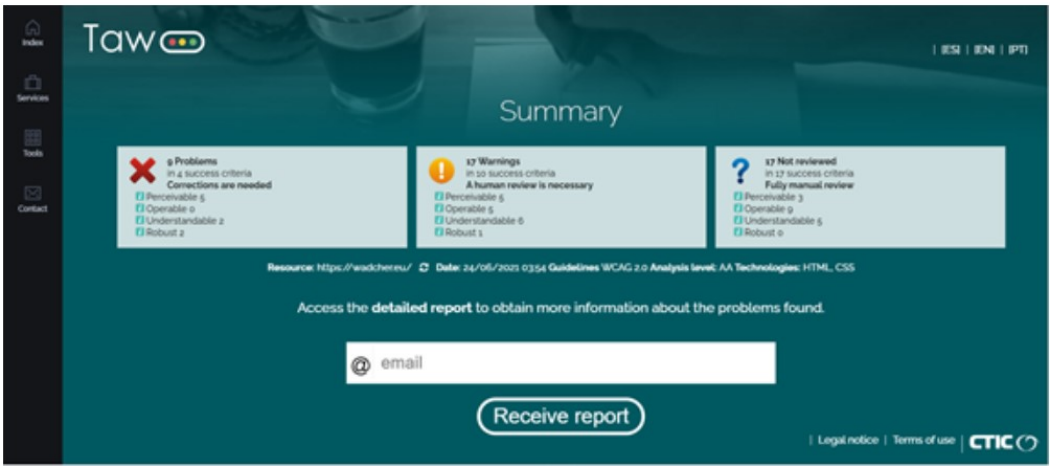

Figure 10:The TAW Tool

4.11 Wave[13]

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.

The "Help" section of the Wave website provides information concerning the evaluation results. There is the documentation for each icon and information boxes used to indicate the accessibility issue on the page during the evaluation process. WAVE claims to detect compliance issues found in the WCAG 2.0 guidelines, WCAG 2.1 guidelines (23 HTML/CSS supported techniques), and many of those in Section 508 (U.S accessibility law). One of the limitations of the public version of WAVE is that it does not allow the user to choose between the W3C Web Content Accessibility Guidelines (WCAG 1.0, 2.0, 2.1) and Priority Level (A/AA/AAA) they need for evaluating the Web page. In the public free version, the Wave evaluation report consists of different areas which allow the user to explore the results at different abstraction levels. At a higher level, it shows a summary view of all the errors. It contains the total number of issues for each of six categories (errors, alerts, features, structural elements, HTML5 and ARIA, and contrast errors). At a lower abstraction level, it shows what type of barriers have been addressed and, for each issue, users can see the meaning of the flagged problem, the solution and also the standards and the reference guidelines in the "references" section.

However, rather than providing a structured technical report, WAVE shows the original Web page with embedded icons and indicators that reveal the accessibility information within the page (Figure 11). It also shows a code panel with the problematic part of the code. Once the Web page evaluation has been finished, WAVE does not offer the possibility to create and download an evaluation report. This could be a problem for those who need to save the various states of the accessibility evaluation process or for those who need to communicate the results to other people involved in the accessibility process.

Through a test consisting in evaluating a Single Page Application website, we were able to infer that WAVE cannot evaluate highly dynamic pages because in this case the preview evaluated page is almost empty (without the

[13] https://wave.webaim.org/

sections dynamically created when the page is loaded on the browser). Finally, it does not provide any accessibility metric.

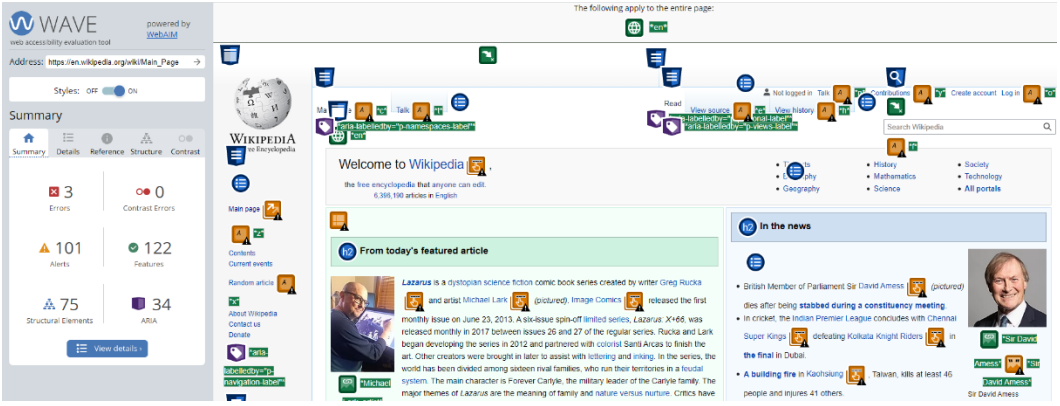

Figure 11: The WAVE Tool

4.12 Overall Analysis of the Considered tools

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, a user 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.

The dynamic support feature is an aspect that is rarely listed among a tool's feature. This feature can be quite useful in the case of Single Page Applications (SPA) or highly dynamic Web pages populated through external services calls. This aspect 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 tools MAUVE++ explicitly states that it is able to support this feature. Also EqualWeb and IBM 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. Also QualWeb supports this feature because it exploits Pupeeter[14], a library that launches a headless version of Chromium that can provide the complete loaded DOM to the validator (such information has been gathered from the library dependencies listed in the git hub repository).

The *Accessibility Percentage* provided by some tools is a metric that summarises how much a Web page/site is accessible. In addition to such metric, MAUVE++ also offers another metric called *Accessibility Completeness*, defined as a metric indicating 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).

[14] https://github.com/puppeteer/puppeteer

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 errors; 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.

Tool name	Guidelines	Supported Techniques	Dynamic Web Page Support	Accessibility Metrics	Result Categories
ACE by accessiBe	WCAG 2.1 AA	No info	No	Accessibility Score & Compliance Level	Success/Failed/Neutral Score
Accessi.org	WCAG 2.0, 2.1	No info	No info	No metrics	Low or Medium impact
Accessibility scanner by UserWay	No Info	No info	No Info	No metrics	Number of tests (passed, failed, not applicable); Violations (low, medium high severity)
Equal Web	WCAG 2.1, ADA, Section 508 and EN 301549	No info	Yes (through browser extension)	No metrics	General errors/Contrast errors/Notices/Warnings /ARIA attributes/ ROLE attributes
EXPERTE	No info	41 features across eight categories (Navigation, Aria, Names&Labels, Contrast, Tables&Lists, Audio&Video, Internationalization & Localization)	No	Accessibility Score	Number of tests (passed, failed, not applicable)
Free Web Accessibility check by AlumniOnline	//	66 accessibility issues	No	No metrics	Errors
IBM Equal Access	WCAG 2.1 (A, AA),	96 requirements	Yes (through	Percentage (Percentage of	Violation/ Needs review/ Recommendation

Accessibility Checker	US 508, EN 301 549		the browser extension)	elements with no detected violations or items to review)	
MAUVE++	WCAG 2.0, 2.1, ACT Rules	107 HTML, 8 CSS	Yes	Accessibility Percentage/Accessibility Completeness	Errors/Warning/Success
QualWeb	WCAG 2.1	43 HTML, 5 CSS	Yes	No metrics	Passed/Failed/Warning/ Not Applicable
TAW	WCAG 2.0 (A, AA, AAA)	No info	No Info	No metrics	Problems, Warnings, Not Reviewed
WAVE	WCAG 2.0, 2.1, Section 508 (U.S accessibility law)	23 HTML,/CSS	Yes (through the browser extension)	No metrics	Errors/Alerts/Features/Structural elements/ HTML5/ARIA/Contrast errors

Table 1: Tool Transparency features

5 THE SURVEY

We conducted a survey in order to gather a better understanding of opinion and expectations of users about the topic of transparency of accessibility validation 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 and (i.e. EU H2020 Wadcher) and channels (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 questions that participants had to answer using a 5-point scale (where 1 was the most negative score and 5 was the most positive one, i.e. 1=not very useful; 5= very useful), and in which only the extremes were explicitly labelled.

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 work, 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 their 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 particular, it asked, on a scale from 1 (=not very useful) to 5 (=very useful), how useful the user rates some 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 (Q1);
    - That the tool specifies how it categorizes evaluation results (errors, warnings, etc.), (Q2);
    - That the tool is able to provide general measures that make explicit the level of accessibility of the website/mobile app (Q3);
    - 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) (Q4);
    - That the tool gives some practical indications on how to resolve the detected problem (Q5);
    - That the tool gives some indication of its limitations, also asking which ones (Q6).
- In addition, a question asked whether users 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; 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: Standards in publishing, digital business, informatics, non-profit organization.	2,88% (in tot)	1 each

Table 2: Work organizations of participants

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 experts	36,23%	50
Web developers	24,64%	34
Web commissioners	10,87%	15
Researchers	5,80%	8
IT managers	3,62%	5
People supporting digital transition of public administrations	2,17%	3
Students, test specialists, academics/scientists, UX/UI experts, 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 the answers that the users provided to the questions included in the survey.

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

90 people answered to 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 use 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 were grouped into different themes (please note that it sometimes happened that, in their responses, users mentioned more than one aspect), reported below by first mentioning the most frequently mentioned ones.

31 users mentioned the information about **errors/violations of accessibility** as an aspect that characterizes the transparency of a tool. In particular, 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; that they offer a good coverage of the errors and they clearly categorize

them i.e. depending on the languages (e.g. HTML, CSS), using relevant references to the page/code to better identify/localize them, and also using relevant references to the corresponding concerned criteria; and finally, that they explain why an issue was pointed out and which are its consequences in terms of accessibility. 24 users mentioned some more general **characteristic/quality that tools should have** in order to be transparent. Among the qualities mentioned by the users there was the tool's easiness of use, clarity, usability and reliability; also, the fact that the tool is free or open source, that it is independent from specific stakeholders, 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.
17 users mentioned that transparency is highly impacted by how clear are the **results** provided by the analysis, and also how comprehensible is the way in which the analysis is carried out. With this category we refer to broader information on the resulting analysis (i.e., not just focusing on the errors), i.e. how the results have been obtained, the pages used for the evaluation, the completeness of the analysis. In particular, under this category, we grouped together people's comments mentioning the clarity and comprehensibility of the results produced by the analysis and the way in which they have been obtained, the clarity of the categorization of the results, the clear identification of the pages used (and those not used) in the evaluation, the completeness of the analysis, its reliability, and verifiability/replicability. 17 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 a factor that affects the transparency of a tool is "when the tool highlights the methods and the parameters that characterize the evaluation criteria". The **standard(s)** with the tool is compliant with have been mentioned by 16 users as a way to characterize the transparency of tools. 15 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. 9 users mentioned **specific features of the tool** that can affect transparency. Among them, 5 users highlighted the users' need of getting 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. 6 persons highlighted that one factor that affects the transparency of a tool is whether the tool highlights the **specific techniques/tests** that the tool carries out (or not) when checking the various success criteria. Another aspect that users (N=5) judged important for the transparency of the tool is that it should clearly highlight the situations that the tool is not able to address automatically and therefore for which situations there is a specific **need of manual checking**. In particular, one user said that one aspect that characterizes the transparency of a tool is when it 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 of manual checking in general (as tools are never exhaustive), but also to provide guidance to the users about what carrying out this manual checking would imply in more concrete terms. 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 explains the choices that it did to come to the provided results. 4 users highlighted that further information about the **methodology** and objectives of the tool should be provided to users, to increase the transparency of the tool. Moreover, the importance of declaring the **limitations** of the assessment provided by tools was mentioned by 4 users as a way to improve transparency. As **further aspects** mentioned by participants (N=3), two of them highlighted that inconsistencies that can be found among the evaluations provided by different

tools can affect transparency. A user mentioned that a tool is transparent when it is actually possible to perform some modifications to the validation results (i.e., when it is possible to declare that a "fail" is actually a "pass").

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? (Q1)
- specifies how it categorizes evaluation results (errors, warnings, etc.)? (Q2)
- is able to provide general measures that make explicit the level of accessibility of the website/mobile app? (Q3)
- presents the evaluation results in a summarized (e.g., graphs, tables) and in a detailed way (e.g., code view)? (Q4)
- gives some practical indications on how to resolve the detected problem? (Q5)
- gives some indication of its limitations? (Q6)

	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

As it can be seen from Table 4, since the median (Mdn) values are higher than mean (M) values the mass of the data in the 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.

For example, on which types of limitations the tool should provide indications?

The answers were grouped into 5 different 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.

36 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 in its structure; when checking

colour contrast on pseudo-elements; when they have to analyze mobile apps, when they have to analyze 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 i.e. 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 ACT published 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 towards specific standards. One user highlighted that there are some tools which are not very up-to-date; Another comment indicates that the versions of the various languages (i.e. JavaScript) and frameworks (i.e. Bootstrap) that the tool is able to address could represent a limitation to the analysis that can be done by the tool, thereby it should be clearly indicated. Another type of limitation regarded the occurrence of **situations in which the results can be wrong (e.g. false positive or false negatives), or ambiguous**. 7 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 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 of manual check (with possible guidance on this manual check).** 5 users emphasized the fact that an automatic validation is never complete/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 (N=7), 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 that they can carry out. Another aspect mentioned by a user is that sometimes tools are too "code-based".

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, lack of clarity.

17 users mentioned difficulties connected with the **results** provided by tools. The aspects that they mentioned regarding the results can be grouped according to *4 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 to experience a mismatch between what they observed and what was reported by the tool, i.e., 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 limit to provide a technical summary which makes it difficult to understand the 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.

16 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 i.e. 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. Finally, one user highlighted that sometimes errors are specified in English, and this could represent a further barrier when trying to understand an error, especially for those that do not have sufficient familiarity with this language. 14 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/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 content or no*t". 11 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.

17 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. 13 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 more precise indication of the conformance level (i.e. A, AA) 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. 5 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 to 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 a 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 positive, 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 is not able to be sure that it is an actual error.

5.4 Effect of Frequency of Use of Tools and Level of Technicality of Users on Answers to Q1-Q6 Questions

In order to calculate whether there was some effect of technicality level and/or familiarity of users with tools on the answers to Q1-Q6 questions, on the one hand, users were divided into 'non-technical people (i.e., people who should refer to a technical-computer expert to solve accessibility problems emerged from automatic validation, and therefore 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, and therefore people with computer skills that concern the development and the writing of code. Regarding the familiarity of users with accessibility validation tools, we rated it in terms of how many times the users declared to use 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). Regarding 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.

In order 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 questions Q1-Q6 we first checked the normality of the concerned distributions. Since all the distributions were non-normal, we ran the non-parametric unpaired two-samples Wilcoxon test, finding a significant effect of the frequency of use of the tools to some of the questions. 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 question Q2. 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

question Q3. 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 particulary 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 Q6. 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 with 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.

We found no significant effect of the level of technicality to all the above questions.

6 USER TEST

In order to have more direct user feedback about tools' transparency, a user test was carried out. In the test, the users had to perform some tasks with three different accessibility validation tools. 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 the small group of tools that meet such requirements, we selected these three because they are available in different ways: one is mainly a stand-alone Web application (Mauve++), one (Lighthouse) is integrated in a 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.

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 was video-recorded with the users' permission. During the test, there were always two moderators that introduced the study's motivations. In order to better follow the test, the users were asked to share their screen with the moderators. 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 balance learning effects on the 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. Before starting the test, users received the links corresponding to the three tools, the link to the page to evaluate using the three tools (https://en.wikipedia.org/wiki/Main_Page/) and the link to the questionnaire. Since the users had to fill in the questionnaire while running the test, in order to prevent any influence on their 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 the following tasks, which were provided through a google Form questionnaire. The questionnaire includes questions regarding some information that the users need to find in the tools as well as subjective feedback on the three tools evaluated. The users had to perform the following tasks:

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

- 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, they had to indicate the supported standards.

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

- How are the results of the assessment of each Web page element classified?
- Did the tool provide you with information explaining the meaning of each category (please also copy-paste the relevant information found)?
- How many errors have you found?
- How many elements does the tool indicate that it is not able to check automatically?
- 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?
- What do these measures describe in your opinion?
- Do you find the presentation of the evaluation results useful and understandable (and motivates the answer)?
- Analyze one of the errors the tool detects. What information is given on the type of problem?
- What information does the tool provide on how to solve the problems identified in the assessment?
- Does this information seem sufficient to you to solve the problem? Explain your answer
- Browsing the tool, do you find a point where the tool states its limitations in carrying out the accessibility assessment?

Task 3. Final considerations on the tool by answering the following questions:

- In terms of transparency, which features of this tool did you like the most?
- In terms of transparency, what aspects of this tool did you dislike?

- In terms of transparency, which features should this tool have that you have not found?

Such tasks have been defined in order to drive the users to testing the different ways to evaluate and produce the validation results in all the tools.

Lastly, they had to rate on a scale from 1 (no transparent) to 5 (fully transparent) the three tools.

6.2 MAUVE

The answers for the MAUVE++ tool were the following. Regarding whether the tool provides information on the guidelines supported, 11 answered yes, 7 answered partially (4 found only standards, 3 did not find the techniques). Regarding how the tool classifies the results of the assessment of each Web page element, and whether it provides information on such classification, 15 users correctly answered to this question by indicating the three types of results provided by the tool (error, warning, success), and 3 users indicated other types of information generated by the tool.

Regarding the number of errors found, 4 users correctly answered these questions, showing to have correctly understood the difference between errors and warnings, while the remaining 14 ones answered this question partially correctly, as they did not show to have understood the category of "warnings" (i.e. the elements that cannot be automatically evaluated by the tool).

When asked how many elements the tool indicates that it is not able to automatically check, the answers were varied: 1 indicated 26%, 6 5%, 6 answered that it was not indicated explicitly, 4 indicated 7, and 1 4%.

Regarding whether the tool provides quantitative indicators of the accessibility and completeness of the evaluation, the answers were all positive.

Regarding the question on whether the presentation of the results is useful and comprehensible, the answers were yes (12), in part (5), no (1). The positive comments appreciated the various ways to present the evaluation results, along with relevant detailed information. Those who were partially positive found some pieces of information provided not immediately understandable, also providing some suggestions for improvements. 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 finally indicate whether it is sufficient to solve the problem. 11 users found the information provided 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, there were mixed answers, 9 positive and 9 negative answers. The positive ones actually indicated various types of information provided by the tool, such as a pie chart indicating the percentage of the checkpoints actually assessed (evaluation completeness), and the file with supported success criteria.

The question regarding what aspects they liked most in the transparency perspective 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, the answers were varied as well: sometimes the error description is unclear, the internal specification of the guidelines is not public, how to find the points with errors can be improved, the possibility to add further filters in the presentation of the results for developers, the difference between errors and warnings is not explained clearly.

6.3 QUALWEB

For the QualWeb tool, at the question regarding whether the tool provides information on the guidelines supported, 16 answered yes, 1 answered partial (only standards found), and 1 answered no. In general, most of them also noticed that it also provides support for the ACT rules. Regarding the presence of the categorization of the accessibility results, 17 answered yes (errors, warnings, passed, not applicable), 1 answered that errors were classified but the categories not indicated.

The question concerning whether such categories are explained in the tool, 13 answered no and 5 yes, however the five who answered positively then indicated pieces of information that were explanations of the techniques checked rather than of the results categories.

When asked about the number of errors detected, 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), and 1 person showed not having understood the categorization at all.

Regarding whether the tool provides overall accessibility indicators 9 answered positively and 9 negatively. Those who were positive indicated the number of errors and checks positively passed as such indicators.

The question whether the results presentation is useful and comprehensible received 12 partly answers, 5 yes and 1 no. In the negative aspects, some users mentioned the lack of showing the errors directly in the Web page user interface and code, the lack of graphical representations of the results, the results are shown without following any clear order and thus it lacks a browsable overall view. In contrast, on the positive side, they mention the initial summary of the assessment results, the filters of the results shown, the possibility to see the code extract associated with the error, sometimes it provides useful information to analyse the errors, the use of the accordion to provide more detail on the errors detected.

Then, users were asked to analyse one of the errors detected and indicate the associated information provided by the tool. Regarding the information on the problem detected, they indicated that the tool provides information on the problem type, associated technique and the excerpt with the page element involved, along with access to the W3C relevant information.

Regarding whether the information for solving the problem detected was found sufficient, ten answered positively, while eight expressed also concerns, such as that the information is suitable only for developers or accessibility experts, and requests for improvements, for example, to better understand how the problem should be solved or to explicitly indicate most important aspects to correct or to show the error in the Web page code.

Regarding whether the tool indicates its limitations, 15 answered negatively, and the remaining three reported that they found some indications such as in the referred W3C documentation, in the indication of the guidelines supported, in the indication of the non-applicable techniques.

Regarding the aspects they liked most from the transparency perspective, users mentioned the number of not applicable assessments and that of those positively performed, the error classification and that for each error it is indicated the violated success criteria and a possible solution, it is possible to filter the results presentation, it provides the excerpt of the code corresponding to the error, it is open source.

Regarding the aspects that they did not like, the users mentioned the lack of the following elements: showing on the user interface the elements that generated accessibility errors, any order in the results 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 a preview of the page with errors highlighted, 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 what ACT rules are, better explain the coverage with respect to WCAG guidelines.

6.4 LIGHTHOUSE

When testing the Lighthouse tool (Figure 12), regarding whether this tool provides information on the guidelines supported, 16 answered that the tool does not provide such info, 2 answered positively, even if only to some extent (ARIA standards were mentioned by both users, the description of the accessibility problems that have been found was mentioned by one user). Actually, it is worth noting that the Lighthouse Chrome's extension does not provide such general info to users upfront (i.e. the ARIA standards can be mentioned only after validating a specific page).

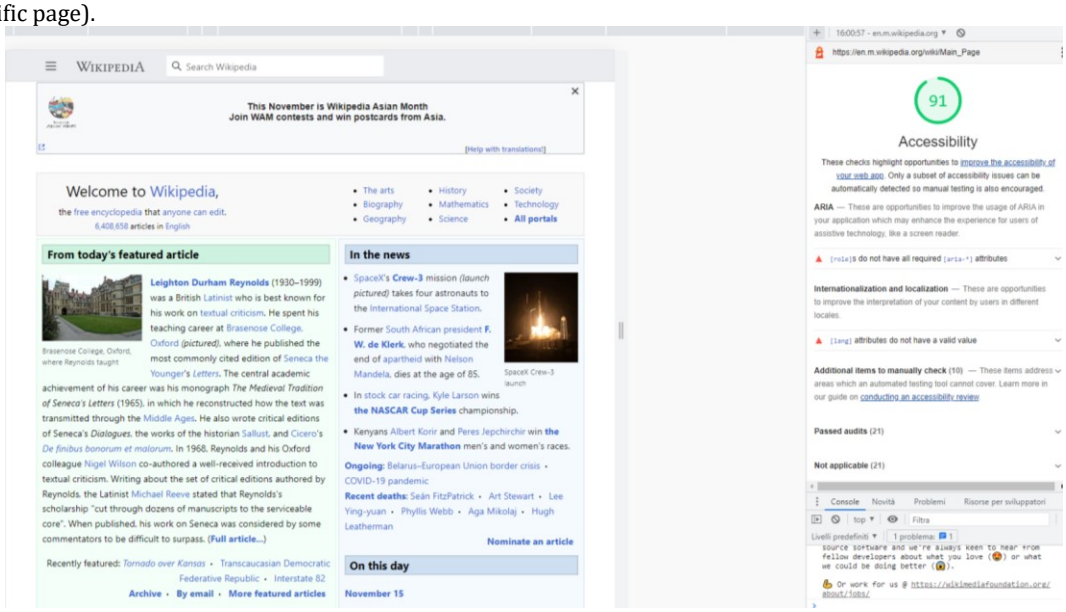

Figure 12: The Lighthouse tool

As for how the tool classifies the results of the assessment and whether it provides information on such classification, 9 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 other categories, depending on the type of problem found (such categories are: ARIA, Navigation, Internationalization and localization). 4 users recognized the categories just in a partial manner, while 5 users did not recognize at all the different ways in which the tool classifies the results.

When asked how many errors the tool identified in the considered Web page and how many elements the tool identified as not able to automatically check (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, all participants answered affirmatively. The majority of them correctly answered that such indicators offer a summative and synthetical measure of the accessibility of the considered page, while a few of them (N=5) expressed some doubts and unclarity on the actual meaning of such indicators and how they were concretely calculated.

As for the question on whether the results presentation is useful and comprehensible, 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 concerned code portions (by clicking on the associated link). Those who were partially positive found that sometimes it is 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 say in which way the checks fail; another user complained that it is not easy to find immediately the total number of errors; another user highlighted the need of grouping 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 the user 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 users were asked to analyze one of the errors detected and indicate the associated information provided by the tool. The majority of users (N=16) correctly found in the tool the information about the errors detected, two users did not. Regarding whether such provided information was sufficient to solve the problem, 7 users found that the tool does not provide sufficient information to solve the errors, i.e. how to operatively act on the page code to solve the errors. 7 users were overall satisfied about this aspect. The remaining 4 users expressed a 'borderline' point of view saying that, while acknowledging that the tool provides some information, this information is sufficient to address just the simplest cases, while in other cases further deepening is needed, especially by unskilled users.

Regarding whether the tool provides clear information on its limitations, 11 users answered yes, the remaining 7 ones 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 evidence should be added.

The question regarding what aspects they liked most in the transparency perspective received various answers, with varied relevance. Among the aspects that users appreciated more, 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 deepening (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 the answers were varied as well. 3 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, another user declared to have not found any possibility to set the conformance level ("A", "AA", "AAA"). 3 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 is not clear which elements generate errors and also the referred criteria are not 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 lack of having the results of the analysis shown directly in the page and within the code. 5 users raised concerns associated with the usability, intuitiveness and the clarity of the tool and the poor usability of the provided visualizations (i.e. the validated points are not clearly categorized). Another user pointed out that the presentation of results done with the accordion-like widgets is very essential. Another user noted that if a user exploits a small monitor, it could be difficult for him 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; another user did not like the presentation of results done using accordion widgets; 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, the participants mentioned: a more detailed description of the various errors and further information for solving them; add in the categorization also the warnings, make explicit the standards and guidelines referred, add further graphical visualizations, add an indication of the settings according to which the analysis has been done, make available a more 'global' view of the code, add further filters according to which a validation runs, make clearer and possibly group together in a new section the errors that should be checked manually, add further references to the code analyzed, add further information on the tool's limitations.

6.5 Other Aspects

Lastly the users were asked to provide an overall rating of the transparency of each tool on a scale 1 (no transparency) to 5 (fully transparent) the three tools. 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 all the tools do not yet fully support 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 seems still not yet able to provide various relevant pieces of information in a clear manner.

We wanted 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 homogeneous (we apply the Levene test, p-value= 0,6775, H0: the variances of the three groups are homogeneous), and there are more than two groups to compare, we applied the Kruskal-Wallis test, which gave a p-value=0,0001 thereby concluding that there are significant differences between the three groups. However, since from the output of the Kruskal-Wallis test we know that there is a significant difference between groups, but we do not know which pairs of groups are different, 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 ($p < 0,05$).

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 further trust 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 the 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. Also in the user test it emerged that users often appreciate the possibilities to receive reports targeted to the various types of possible uses, as simply providing information may be insufficient, as it is often necessary to consider the likely audience that will receive this information. Thus, information should be provided in a way to be correctly interpreted and understood by the audience and able to consider 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. Moreover, reports should be interactive, with the possibility of filtering the results (i.e. to see only the info 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, for example it was not clear which element generated the error (as accessibility issues are not indicated within the relative code). 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*, 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 tools' 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 high-level descriptions and sometimes tool developers can interpret them 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[15] 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.

[15] https://www.w3.org/TR/act-rules-format/

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 just 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 general reflection concerns that the validation of accessibility guidelines is becoming more and more complex. The WCAG 2.1 have 82 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.

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. 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. Also the user test showed 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 of the main results that came out is that transparency should not consist in providing exhaustive information about how the tool works in all its details (which sometimes can generate confusion rather than providing clarifications), rather, that the information provided by tools should be easy to understand and, especially, match the needs of different stakeholders (who could 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.

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

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

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

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

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

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

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

Beirekdar A., Keita M., Noirhomme M., Randolet F., Vanderdonckt J., Mariage C. (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

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

Brajnik, G., Yesilada, Y., & Harper, S. (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.

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

Andreas Burkard, Gottfried Zimmermann, and Bettina Schwarzer. 2021. Monitoring Systems for Checking Websites on Accessibility. Frontiers in Computer Science 3 (2021), 2.

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

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

Tânia Frazão and Carlos 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

Fuertes, J. L., González, R., Gutiérrez, E., & Martínez, L. (2009). Hera-FFX: a Firefox add-on for semi-automatic Web accessibility evaluation. Proceedings of the 2009 International Cross-Disciplinary Conference on Web Accessibililty (W4A) (pp. 26-35). ACM.Giovanna Broccia, Marco Manca, Fabio Paternò, and Francesca Pulina. 2020. Flexible automatic support for Web accessibility validation. Proceedings of the ACM on Human-Computer Interaction 4, EICS (2020), 1–24.

Greg Gay and Cindy 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.

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

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

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

Lazar, J., & Olalere, A. (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.

Aliaksei Miniukovich, Michele Scaltritti, Simone Sulpizio, and Antonella 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

Mirri, S., Muratori, L. A., & Salomoni, P. (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.

Lourdes Moreno, Rodrigo Alarcon, Isabel Segura-Bedmar, and Paloma 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

Ashli M Molinero, Frederick 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.

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

Nietzio, A., Eibegger, M., Goodwin, M., & Snaprud, M. (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

Marian Pădure and Costin 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

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

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

Petrie, H., King, N., Velasco, C., Gappa, H., Nordbrock, G.: 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

Power, C., Freire, A., Petrie, H., & Swallow, D. (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.

Schiavone A., Paternò F., 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.

Navid Salehnamadi, Abdulaziz Alshayban, Jun-Wei Lin, Iftekhar Ahmed, Stacy Branham, and Sam 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

Q. Vera Liao, Daniel M Gruen, Sarah Miller, Questioning the AI: Informing Design Practices for Explainable AI User Experiences, CHI 2020

Markel Vigo, Justin Brown, and Vivienne 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.

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

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