Rapid Review: Automated Testing for Website Accessibility

Nils Larsson

2025-04-03

1. Introduction **2**
1.1 Background and Context 2
1.3 Research Questions 2
2. Methodology **2**
2.1 Problem Formulation 2
2.2 Search Strategy 3
3. Results **4**
RQ1: What is possible to test and how effective is automated testing? 5
RQ2: What types of automatic accessibility testing tools are there? 5
RQ3: What are common best practices of using automatic testing tools? 6
4. Discussion and Threats to Validity **7**
4.1 Implications of Findings 7
5. Evidence Briefing **9**
6. Conclusion **10**
References **11**
Appendix **13**

1. Introduction

1.1 Background and Context

In website development, accessibility is an area which concerns the accessibility for various groups of users with various kinds of impairments. In some cases, accessibility is mandated by law, whereas in others, accessibility is important in order to make a commercial product available to a larger group of users.

Accessibility testing is an activity that often occurs in the development process, in order to ensure that accessibility guidelines are being followed and that the websites will become accessible. It can also be an activity that occurs to evaluate a website in a later stage, as websites need continuous development.

There are certain problems with maintaining accessibility on websites. Even on small websites with only a few subpages, manual inspection might not be enough. If the website is larger and contains perhaps many subpages, manual testing can be a challenge and an overwhelming task. Some sort of automation can then be a helpful tool.

A global standard for accessibility testing is the WCAG guidelines, introduced by WC3 [1]. There are a number of success criteria which need to be fulfilled in order for a website to be deemed accessible. The automated test tools for accessibility are based on these success criterias, but their coverage varies.

1.2 Purpose and Objectives

In this Rapid Review, the purpose is to look at how accessibility can be achieved when developing and testing websites, so that they comply with common standards (WCAG), using automated accessibility testing tools.

The objective is to understand how websites can achieve accessibility in the developing and testing process using automatic tools. And by gaining this understanding, give some actionable recommendations, when using automatic testing tools for increased accessibility.

1.3 Research Questions

The following research questions was chosen:

- RQ1: What is possible to test and how effective is automated testing?
- RQ2: What types of automatic accessibility testing tools are there?
- RQ3: What are common best practices of using automatic testing tools?

2. Methodology

2.1 Problem Formulation

How accessibility testing is conducted is multifaceted but a wide array of automation tools for testing and evaluation are available. The aim of accessibility testing for websites is to make the websites more accessible to many users that have some sort of impairment.

By taking these impairments into consideration, certain features of a website need to be constructed in a way, so that even if you have a certain impairment, you will be able to access the particular website.

In the WCAG guidelines there are four principles, called “POUR”: Perceivable, Operable, Understandable and Robust. Under each principle there are guidelines, and under each guideline there are success criterias [2].

The tools used in automated testing are based on these guidelines and success criteria but how they implement the ruleset may vary. There can therefore be differences in the tools how effective they are.

Automated testing can have an appeal when it comes to testing, because it can give the promise of speeding up the process and being a simple and universal solution for accessibility testing. But is this really true? In this rapid review, the area of automated testing for accessibility on websites is being considered.

2.2 Search Strategy

Databases and Search Strings

The selection of the database was Scopus as it covers relevant areas of computer science. Being a rapid review, it was deemed sufficient to limit it to one database, as described by Cartaxo et al. [3]. The search was made 2024-12-24.

Search string

"web*" AND "accessibility" AND "test*" AND "eval*" AND "auto*" AND "tool*" AND "WCAG"

AAK (article title, abstracts, keywords): 75

Inclusion and Exclusion Criteria

Inclusion: Peer-reviewed material, “computer science” + “engineering” + keyword: “websites”, articles, conference papers

Exclusion: non English

Selection procedure

Selection procedure	# of papers
Papers extracted from database:	75
After applying database inclusion criterias: “Computer science” + “Engineering”:	67
Then peer-reviewed: articles and conference papers:	64
Then limiting to “websites” by keyword:	45
After applying database exclusion criterias: non-English:	44
After screening:	17

The papers were then downloaded in PDF format. Then by manual inspection and going through AAK, for 7 of the papers, full-text was missing, 2 were deemed potentially out of scope, 16 papers were deemed out of scope for various reasons and 17 papers remained after the first screening.

After the immersion process, by reading the papers, 10 papers that initially were deemed in scope were deemed out of scope and 7 of the papers that were first deemed as out of scope were re-included. None of the papers deemed potentially out of scope were included. After a more thorough search, 1 of the missing full-text papers was reincluded.

In total, 17 papers were remaining after the selection after immersion.

Extraction procedure

Using thematic analysis, the first step was to read the full-text studies (immersion) as described by Cruzes and Dybå in [4]. While reading potential codes were written down as initial codes. The research questions were also helpful in the coding process.

The papers were then coded using an open source PDF software named Okular where highlights and notes could be added to the PDF files (annotations, or codes). The annotations or codes could then be extracted using a NodeJS app called pdfannots. By using pdfannots, highlighted keywords or sentences or notes could be extracted. They were extracted to JSON, which could be converted to CSV using a Python script, for a tabular view.

From the collected CSV file, themes could be developed as well as groupings that belonged to a particular research question.

Synthesis procedure

Each paper of PDF format was in this way coded. After the coding and the extraction, the collected codes could be organised into themes and higher order themes as described by Cruzes and Dybå in [4]. The hierarchy could then be viewed as:

code (keyword or sentence) → theme → higher order-theme

Higher-order themes would correlate with the research questions and themes would correlate with subquestions. For example in RQ3: “Best practices” (higher-order theme) could be further divided into sub questions and themes of “advantages” and “disadvantages” and “best practices in general”.

Notes were also taken separately in a text document, where initial remarks, findings and some notes were recorded. Included in the codes were also key findings, which were shorter summaries of the papers, taken from the abstracts and the results of the papers. From the papers and the coded material, there were also certain statistics gathered and summarised. Finally, in a spreadsheet, each paper was recorded along with information and the themes for a better overview.

Reporting

Using the summarised data and the recorded themes it was possible to create the figures and write the review, where each study was numbered and referred to, along with findings and themes.

The themes and findings of the papers were then related to the research questions in the results section and discussed and correlated with the research questions in the discussion section of the rapid review as described by Cruzes and Dybå in [4].

The results and recommendations were also included in the evidence briefing section. Finally key insights from the results and discussions were included in the conclusion.

3. Results

The selected 17 studies could be grouped according to certain types of categories. While all related to automated accessibility testing, they differed in various ways. See Table 1 in the appendix for a full list of the studies and their key findings.

Some could be grouped as relating to automated testing focusing on automatic tools in

general and metrics and some could be grouped as a more applied type of studies.

As to the group that were more grouped to automated testing in general, some were specific on a metric: [5], [6], evaluating the tools themselves: [7], [8], [9], [2], [10], and some were specific on a particular technology: LLM: [11], or evaluating accessibility using an own developed tool: [6], [12], or methodology: [13].

One group of studies was a more applied type of studies where accessibility was one part of website evaluation, and the other part was usability, and then other things like SEO compatibility and performance etcetera could be evaluated: [14], [15], [16], [17], [18], [19] and [20].

RQ1: What is possible to test and how effective is automated testing?

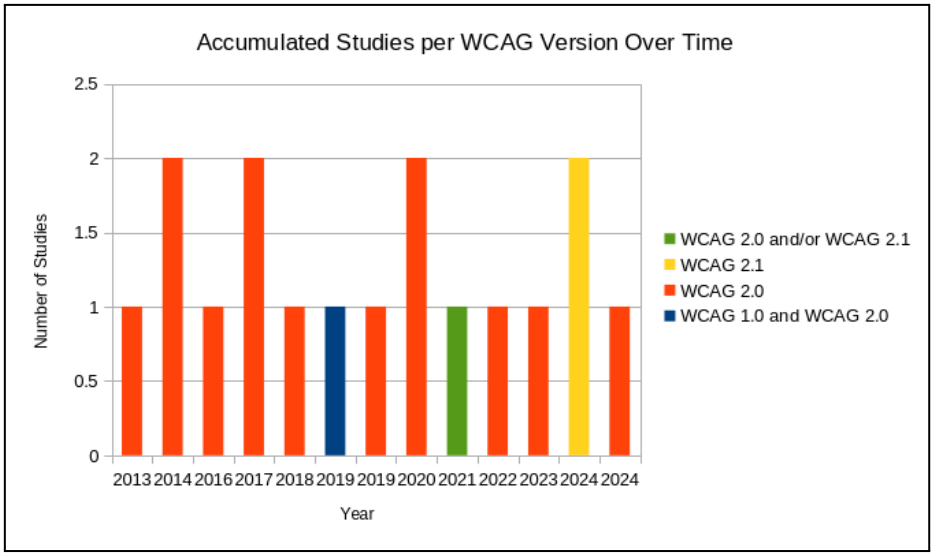


Fig. 1. The different WCAG versions in the studies, as accumulated number of studies per WCAG version over time (year).

Initially it can be said that the automated testing in studies was based on the WCAG guidelines as it is a global standard. In the majority of studies, WCAG 2.0 was the main guideline version. However, WCAG 1.0 and WCAG 2.1 were also tested against. See figure 1 in appendix for further information.

Measuring Accessibility Issues: Coverage and other Metrics

Generally the metric or measurement used is counting how many success criterias a specific tool uses in a WCAG guideline version divided by the total number of success criterias in the WCAG guideline version, giving the coverage in percentage. Most tools also report on the total amount of issues and what type of issue it is. However, it should be noted that different types of metrics were used in the studies as well as in the tools.

The term coverage was used in different ways in the studies. Generally, coverage related to how many success criteria a specific tool reported at least one error on as described by Vigo et al. [9].

As the studies indicate, the percentages of coverage vary, and it can be difficult to find an exact percentage but from 23% up to 50% coverage was reported in [9], where also three studies ranging from 44-55% was referenced. Another researcher estimated the coverage to

about 50% in [7].

There are also other metrics or measurements used in the studies, including metrics that the researchers created themselves [5], [6].

Completeness and correctness was also used as metrics. Completeness according to Vigo et al. [8] meant using an expert group's reporting as a baseline and then comparing the tool's reported errors with the baseline or actual errors. The completeness score in [9] was between 14% and 38%, meaning that of the 50% covered success criteria only between 14% and 38% of these are actual errors, depending on which tool is being used.

Correctness measures how many false positives have been reported and how many of these are false positives. Some tools report on errors that are in fact not errors, and the more they do so, the less correct they are. Many tools have high correctness but two tools that had lower correctness were TAW and TotalValidator as noted in [9].

RQ2: What types of automatic accessibility testing tools are there?

Classification of Tools

The types of accessibility testing tools can be categorized or classified in various ways. One way, described in [8] is classification by platform: online service, within a browser, within an authoring tool or install on hard drive.

Relevant for this rapid review is mainly test tools and their subcategories: general, specialized and service. General test tools were the tools that were mainly used in the studies and in the case studies. These are tools that are aimed to find many different types of barriers ([8]). Specialized tools are tools that aim at a particular barrier. Tools classified as service, are tools that continuously monitor websites [8].

WC3 lists many accessibility testing tools [21] and it was referenced in many of the studies, for example in [10]. In two cases own developed tools were used: [6], [12].

Specific tools

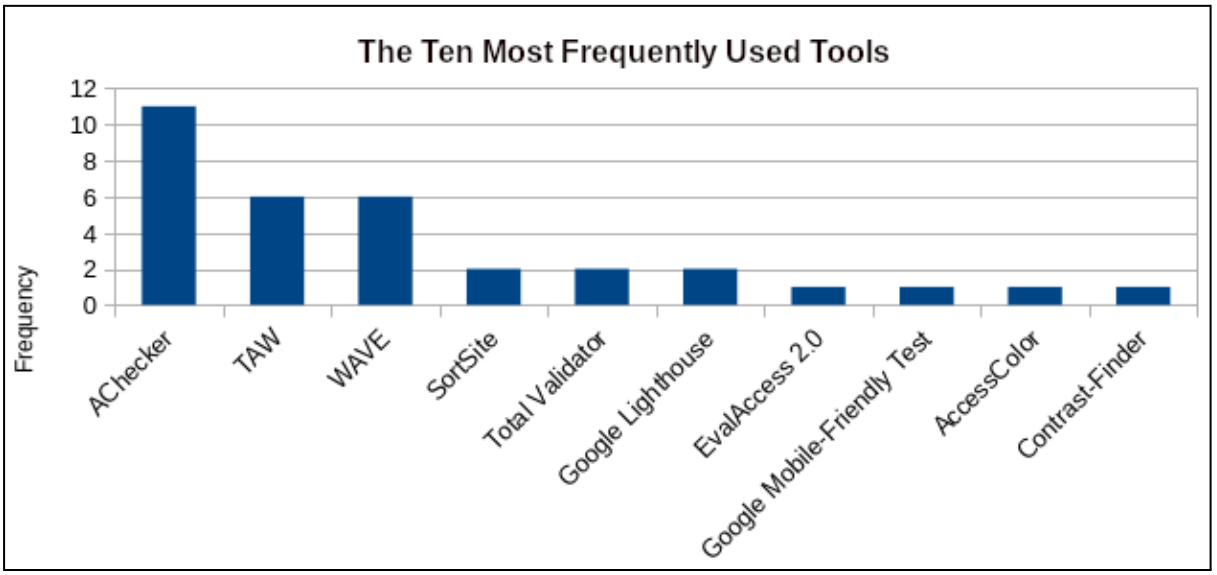


Fig. 2. The ten most frequently used tools in the studies. See appendix for a figure of all studies.

It was noted that some of the tools were used more frequently. Achecker was the top most used tool, used in 64.7% of the studies. Thereafter was TAW and WAVE, used in 35.3% of

the studies, in a tied second place. In a tied third place were SortSite, Total Validator and Google Lighthouse, used in 11.8%. See figure 2 in appendix for further information.

RQ3: What are common best practices of using automatic testing tools?

Advantages

Advantages of automated testing is the ability to test large numbers of pages in a short amount of time, as mentioned in [10], [9].

Another advantage is with automated testing, potential issues can be found, and the tester or developer can learn more about the issues, and the tools will often provide additional information and links to more information, as was mentioned in [10].

Although it can be difficult to test for certain success criterias, some tools can flag these potential issues for manual inspection [12] and [13].

Disadvantages

WCAG is interpretive in nature, therefore some types of success criterias can not be tested automatically, which was pointed out in [9]. An example of this is the alternative text attribute in image tags which could be a false positive. The alternative text attribute is there, but the description can be incorrect and it would still flag this as a pass.

Similar types of issues were also discussed: for example incorrect link description and incorrect language tag, which are also interpretive. These and other examples were discussed in some of the studies: [7], [8], [6], [12], [11].

One of the studies [11] based on LLM for accessibility testing used LLM technology in testing for various content related accessibility issues that usually requires manual inspection. It showed promise in being able to automatically test also for content related accessibility issues.

Best practices

Due to the fact that different tools cover different success criteria, the tools may complement each other. Using a combination of tools will therefore yield better results than only using a single tool, as described in [2] and [10].

Another best practice was to not solely rely on automated testing which was mentioned in [9]. A similar conclusion is reached in [2] and [10]; automated testing can not discover all accessibility issues and will need to be complemented with manual interpretation and testing.

Consider how accessibility testing is incorporated in the development cycle. Accessibility needs to be addressed in the web development process, but also after development.

To maintain accessibility, automated testing can be performed, for example by continuous monitoring (periodic scans) as described in [18].

4. Discussion and Threats to Validity

4.1 Implications of Findings

Regarding RQ1 a delay in following a WCAG version could thus be seen. As the studies were conducted in the period 2013-2024, it would perhaps be expected that higher WCAG versions would be used (WCAG 2.1 and WCAG 2.2). Since WCAG 2.1 and WCAG 2.2 are

backward compatible, they would include all success criteria from WCAG 2.0, and have added some new success criteria. However, this may indicate that WCAG 2.0 is somewhat of an accepted standard, even though WCAG 2.1 was observed in two of the later studies. In terms of increasing accessibility, higher WCAG versions would be welcomed as that would mean that more success criterias potentially would be covered.

The reason for a quite low highest coverage percentage (50%) seems to be due to the nature of the WCAG guidelines, which are somewhat interpretative. Some success criteria will therefore be difficult to cover using automatic tools. However, LLM technology shows promise in solving some of these issues, and increasing the coverage.

With regards to RQ2 and the tools used and the observed most frequently used tools it can be said that some tools seem to be popular among testers. Being the most used tools does not however need to indicate that they are the best tools. It was noted that two of the popular tools had lower correctness scores, which may indicate that even though a certain tool is popular, it does not mean that it is the most accurate.

What could be observed in RQ3 was that there were certain statements and knowledge that most of the papers shared: that automated testing was not sufficient for accessibility testing.

Some of the other best practices were conclusions of certain studies, while others would include such experience based assessments in various parts of their studies. It would often follow from their individual case studies, which gave insight into how various automatic testing tools were used and how their results could be interpreted.

When it comes to the best practices it was somewhat challenging to classify or categorise the best practices, as it was a more interpretive type of data. Creating the categories was somewhat intuitive but when adding evidence to the categories, some of the findings were not easy to categorise in only one category per finding. Some of the best practices could be viewed both as advantageous and disadvantageous for example.

Some of the best practices did however relate more to accessibility testing in general and not only on automated testing.

Some of the best practices related also more to methods and processes, which was partially part of the scope of this review, but since it was only a part of the scope, it was not explored in more depth.

4.2 Limitations and Threats to Validity

Although accessibility is an issue on other platforms and technologies than websites, due to the limitation of this review website technology was chosen as a focus area since it is a common platform that is available for many devices and can be considered one of the most important platforms. Despite competing technologies such as mobile and desktop applications, websites are still one the most important information platforms.

As mentioned in the discussion, the majority of papers used WCAG 2.0, which by now could be considered a bit dated since newer versions are available. However, since only a few more success criteria have been added and since the search used the term “WCAG” and looked for all papers using any WCAG version and automated testing, more papers using later WCAG versions were not found. In [12] it was difficult to interpret whether it used WCAG 2.0 or WCAG 2.1.

Another point regarding comparison of metrics between papers was that since the tools

often reported on errors differently, some of the results needed interpretation and it was somewhat difficult to generalise the results across different studies. Yet, it was possible to do some generalisation and find some common metrics which were useful in providing evidence based recommendations.

The papers date from about 2013 to 2024 and the earliest papers may to some extent be outdated due to the development in the area. This means that technology used and the situation described at the time may be different today. However, what may still be valid is the approaches and the best practices. Despite there being new technology, many fundamental web technologies are still the same, as well as the accessibility issues. Therefore, the earlier paper still has relevance in the theory that is described but also some of their practices, even though some of their results and the tools used, may not be relevant today.

A question of validity can be raised regarding the papers that included particular case studies. Although they were not specifically addressing automated testing, they all used automated accessibility testing tools and addressed the topic of automated accessibility testing. They also covered some interesting findings, knowledge and best practices that would be of relevance to developers and testers.

They were still relevant from an empirical view but more limited in scope. However, their approach can therefore be seen as a more average type of testing, and how they used the tools and what metrics they chose can be seen in a more practical approach, which was also relevant to this rapid review.

5. Evidence Briefing

Automated Testing for Website Accessibility

Summary and purpose

This briefing reports evidence on automated testing for website accessibility.

Key Findings

Finding 1: Many websites today have issues with accessibility or do not follow WCAG guidelines.

Finding 2: Automated testing can be helpful to increase accessibility when applied in the correct way.

Finding 3: Relying solely on automated testing will not completely resolve all accessibility issues.

Relevance

This briefing is relevant for developers, testers and stakeholders that are incorporated in the development and maintenance processes of websites.

Recommendations

- Make sure to look for a WCAG guideline version that is relevant and is at or above WCAG 2.0. Ideally the newer the better. Currently WCAG 2.1, WCAG 2.2 while waiting for WCAG 3.0.
- Be aware that you may only cover up to 50% of accessibility success criterias using automatic tools.
- Be aware that the actual number errors may be lower (completeness), depending on the tool that is being used. Between 14% and 38% of completeness was reported in some popular tools.
- Use tools that are recommended by WC3 and used by many.
- Do not rely on only one tool. Combine tools.
- Look for tools that are suitable for the type of accessibility issues you are looking for.
- Make sure to check for accessibility issues on a regular basis and not only during development and implement a process to make sure accessibility is maintained throughout the lifespan of the website.
- And ultimately, do not solely rely on automatic testing tools but look at other approaches for increasing accessibility, such as expert testing, user testing and possibly using LLM technology.

6. Conclusion

Automated testing will likely continue to be a trusted approach in achieving accessibility even though in some studies it was shown that its coverage was low. However, even though the coverage was low, in the majority of accessibility studies where automated tools were used in practice and at a minimum of coverage level they were able to detect lots of accessibility issues on many typical and standard websites. This indicates that accessibility is an issue that will continue to be important and need to be addressed using various approaches of testing.

Although the scope of this study was automated testing it became clear that other approaches were deemed equally, if not more, important as automated testing. A combined or mixed approach was often proposed as the most successful.

This had some impacts on the study since the assumption that automated testing might be enough for achieving accessibility or at least that automatic tools would cover the majority of success criteria, was not true. It also became apparent that there were some aspects of interpretation involved in accessibility testing, and especially certain success criteria in WCAG.

The interpretative nature of WCAG lends itself to some typical problems that standard automated testing could not handle very well, and would most often require manual inspection. Therefore the idea of the semi-manual approach, would follow from the initial idea of a fully automated approach.

To some extent, the interpretive nature of the WCAG also mirrors the interpretive nature of the topic itself: namely, that accessibility can be interpretative in nature. However, WCAG can be seen as at least a way of addressing the issue and generally by following the guidelines, or attempting to follow the guidelines, will yield increased accessibility.

New LLM technology shows promise of solving some of the interpretative issues in the WCAG and accessibility in general, but as for now, it has not been widely adopted and established.

6.1 Future Research Directions

The field of accessibility testing is an ever-evolving field that will be relevant as long as websites are being created and used. Which is a long time in the foreseeable future.

As was seen in this rapid review, only using automated testing was not deemed the most efficient when it comes to accessibility testing. A future research direction could therefore be to explore how semi automated testing methods, using both automated testing and manual user testing, are being used when testing for accessibility.

It would also be possible to do a similar case study as was done in some of the studies but in a new area, evaluating accessibility requirements using automatic testing tools and then compare the results to the experiences of certain user groups: developers, expert testers or end users.

It would also be interesting to use some tools and frameworks and test some of the accessibility testing resources that were mentioned in the studies, for example the test suite in [7], or the WC3 test suite since benchmarking of tools is an ongoing concern and especially now when there are new interesting options available, for example LLM:s.

References

[1] W. W. A. Initiative (WAI), “WCAG 2 Overview,” Web Accessibility Initiative (WAI). Accessed: Feb. 25, 2025. [Online]. Available: https://www.w3.org/WAI/standards-guidelines/wcag/

[2] A. Lempola, T. Poranen, and Z. Zhang, “Comparing automatic accessibility testing tools,” in *CEUR Workshop Proceedings, Annual Doctoral Symposium of Computer Science*, vol. 3776. Rheinisch-Westfaelische Technische Hochschule Aachen * Lehrstuhl Informatik V, 2024.

[3] B. Cartaxo, G. Pinto, and S. Soares, “Rapid Reviews in Software Engineering,” Mar. 22, 2020, *arXiv*: arXiv:2003.10006. doi: 10.48550/arXiv.2003.10006.

[4] D. S. Cruzes and T. Dybå, “Recommended steps for thematic synthesis in software engineering,” in *Proceedings of the 2011 International Symposium on Empirical Software Engineering and Measurement*. Banff, AB, Canada: IEEE, 2011, pp. 275–284, doi: 10.1109/ESEM.2011.36.

[5] J.-C. Dubois, Y. Le Gall, and A. Martin, “Designing a Belief Function-Based Accessibility Indicator to Improve Web Browsing for Disabled People,” in Belief Functions: Theory and Applications, F. Cuzzolin, Ed., Lecture Notes in Computer Science, vol. 8764, Cham: Springer International Publishing, 2014, pp. 134–142. doi: 10.1007/978-3-319-11191-9_15.

[6] A. Nietzio, M. Eibegger, M. Goodwin, and M. Snaprud, “Following the WCAG 2.0 Techniques: Experiences from Designing a WCAG 2.0 Checking Tool,” in *Proceedings of Computers Helping People with Special Needs*, K. Miesenberger, A. Karshmer, P. Penaz, and W. Zagler, Eds., in *Lecture Notes in Computer Science*, vol. 7382., Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 417–424. doi: 10.1007/978-3-642-31522-0_63.

[7] M. Tollefsen and T. Ausland, “A practitioner’s approach to using wcag evaluation tools,” in *Proceedings of the 2017 6th International Conference on Information and Communication Technology and Accessibility (ICTA)*. IEEE, 2017, pp. 1–5. doi: 10.1109/ICTA.2017.8336047.

[8] K. Wille, C. Wille, and R. Dumke, “A Test Procedure for Checking the WCAG 2.0 Guidelines,” in *Proceedings of the Universal Access in Human-Computer Interaction. Methods, Techniques, and Best Practices*, M. Antona and C. Stephanidis, Eds., in *Lecture Notes in Computer Science*, vol. 9737., Cham: Springer International Publishing, 2016, pp. 120–131. doi: 10.1007/978-3-319-40250-5_12.

[9] M. Vigo, J. Brown, and V. Conway, “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*. Association for Computing Machinery, 2013, pp. 1–10. doi: 10.1145/2461121.2461124.

[10] R. Ismailova and Y. Inal, “Comparison of Online Accessibility Evaluation Tools: An Analysis of Tool Effectiveness,” *IEEE Access*, vol. 10, pp. 58233–58239, 2022, doi: 10.1109/ACCESS.2022.3179375.

[11] J.-M. López-Gil and J. Pereira, “Turning manual web accessibility success criteria into automatic: an LLM-based approach,” *Universal Access in the Information Society*, Mar.

2024, doi: 10.1007/s10209-024-01108-z.
[12] A. N. Al Jabor, F. Adnan, M. Park, and A. Othman, “Mada Web Accessibility Monitor Tool,” in *Proceedings of the 2021 8th International Conference on ICT & Accessibility (ICTA)*, Tunis, Tunisia: IEEE, Dec. 2021, pp. 1–5. doi: 10.1109/ICTA54582.2021.9809423.
[13] P. Acosta-Vargas, S. Luján-Mora, T. Acosta, and L. Salvador-Ullauri, “Toward a Combined Method for Evaluation of Web Accessibility,” in *Proceedings of the International Conference on Information Technology & Systems (ICITS 2018)*, Á. Rocha and T. Guarda, Eds., in *Advances in Intelligent Systems and Computing*, vol. 721. , Cham: Springer International Publishing, 2018, pp. 602–613. doi: 10.1007/978-3-319-73450-7_57.
[14] S. Paul and S. Das, “Accessibility and usability analysis of Indian e-government websites,” *Universal Access in the Information Society*, vol. 19, no. 4, pp. 949–957, Nov. 2020, doi: 10.1007/s10209-019-00704-8.
[15] R. Ismailova and Y. Inal, “Accessibility evaluation of top university websites: a comparative study of Kyrgyzstan, Azerbaijan, Kazakhstan and Turkey,” *Universal Access in the Information Society*, vol. 17, no. 2, pp. 437–445, Jun. 2018, doi: 10.1007/s10209-017-0541-0.
[16] Ş. S. Macakoğlu and S. Peker, “Accessibility evaluation of university hospital websites in Turkey,” *Universal Access in the Information Society*, vol. 22, no. 3, pp. 1085–1093, Aug. 2023, doi: 10.1007/s10209-022-00886-8.
[17] N. Aloboud, R. Alotaibi, and A. Alqahtani, “Evaluating the Usability and the Accessibility of Saudi E-Government Websites,” in *Human-Computer Interaction. Design and User Experience*, M. Kurosu, Ed., in *Lecture Notes in Computer Science*, vol. 12181., Cham: Springer International Publishing, 2020, pp. 363–372. doi: 10.1007/978-3-030-49059-1_26.
[18] M. Fakrudeen, “Evaluation of the accessibility and usability of university websites: a comparative study of the Gulf region,” *Universal Access in the Information Society*, Oct. 2024, doi: 10.1007/s10209-024-01160-9.
[19] S. A. Adepoju, A. Ekundayo, A. O. Ojerinde, and R. Ahmed, “Usability and Accessibility Evaluation of Nigerian Mobile Network Operators’ Websites,” in *Proceedings of the 2019 2nd International Conference of the IEEE Nigeria Computer Chapter (NigeriaComputConf)*, Zaria, Nigeria: IEEE, Oct. 2019, pp. 1–7. doi: 10.1109/NigeriaComputConf45974.2019.8949622.
[20] K. Fatima, N. Z. Bawany, and M. Bukhari, “Usability and Accessibility Evaluation of Banking Websites,” in *Proceedings of the 2020 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, Depok, Indonesia: IEEE, Oct. 2020, pp. 247–256. doi: 10.1109/ICACSIS51025.2020.9263083.
[21] W. W. A. Initiative (WAI), “Web Accessibility Evaluation Tools List,” Web Accessibility Initiative (WAI). Accessed: Feb. 25, 2025. [Online]. Available: https://www.w3.org/WAI/test-evaluate/tools/list/

Appendix

Protocol

Research question

The following research questions will be evaluated:

- RQ1: What is possible to test and how effective is automated testing?
- RQ2: What types of automatic accessibility testing tools are there?
- RQ3: What are common best practices of using automatic testing tools?

Search strategy

Scopus, 2024-12-24

Search string:

"web*" AND "accessibility" AND "test*" AND "eval*" AND "auto*" AND "tool*" AND "WCAG"

AAK (article title, abstracts, keywords): 75

Inclusion/exclusion criterias

Inclusion: Peer-reviewed material, “computer science” + “engineering” + keyword: “websites”, articles, conference papers

Exclusion: non English

Selection procedure

After applying database inclusion criterias: “Computer science” + “Engineering”

“Engineering” overlapped with “computer science” to some extent: without engineering: 64, with engineering: 67

Then “peer-reviewed: articles and conference papers”: 64

Then limiting to “websites” by keyword: 45

After applying database exclusion criterias: “English” : 44

Then by manual inspection and going through AAK, for 7 of the papers, full-text was missing, 2 were deemed “potentially out of scope”, 16 papers were deemed out of scope for various reasons (too specific on a technology, non relevant technology, not focused on testing, and a group that were specific on a particular type of website were deemed out of scope but may be re-included after full-text immersion of the remaining papers), and 19 papers remained after the first screening.

After the immersion process, by reading the papers, none of the papers that was deemed potentially out of scope was included. 10 papers that initially was deemed in scope was deemed out of scope (too specific on a particular technology, focusing on processes, focusing

on remedying accessibility and not testing for accessibility, being obsolete and one SLR), and 7 of the papers that was first deemed as out of scope was re-included (case studies).

After a more thorough search, 3 of the missing full-text papers were found and re-included. However, one paper was out of scope and two of the studies were basically duplicates, whereas one of the studies was kept, and the second one removed.

In total, 17 papers were remaining after the selection after immersion.

The selection process can be summarized as:

From AAK 44 papers, 27 papers were subtracted after being deemed out of scope or obsolete.

Extraction procedure

Using thematic analysis, the first step was to read the full-text studies (immersion), and through the reading start to look for potential codes, and write down these initial codes. The research questions were also of help in the coding process.

The papers were then coded using an open source pdf software named Okular where highlights and notes could be added to the pdf files (annotations). The annotations could then be extracted using a NodeJS app called pdfannots. By using pdfannots, highlighted keywords or sentences or notes could be extracted. They were extracted to JSON, which could be converted to CSV, for a tabular view."

Synthesis procedure

Each paper of pdf format was in this way coded. After the coding and the extraction, the collected codes could be organised into themes and higher order themes. The hierarchy could then be viewed as:

code (keyword or sentence) → theme → higher order-theme

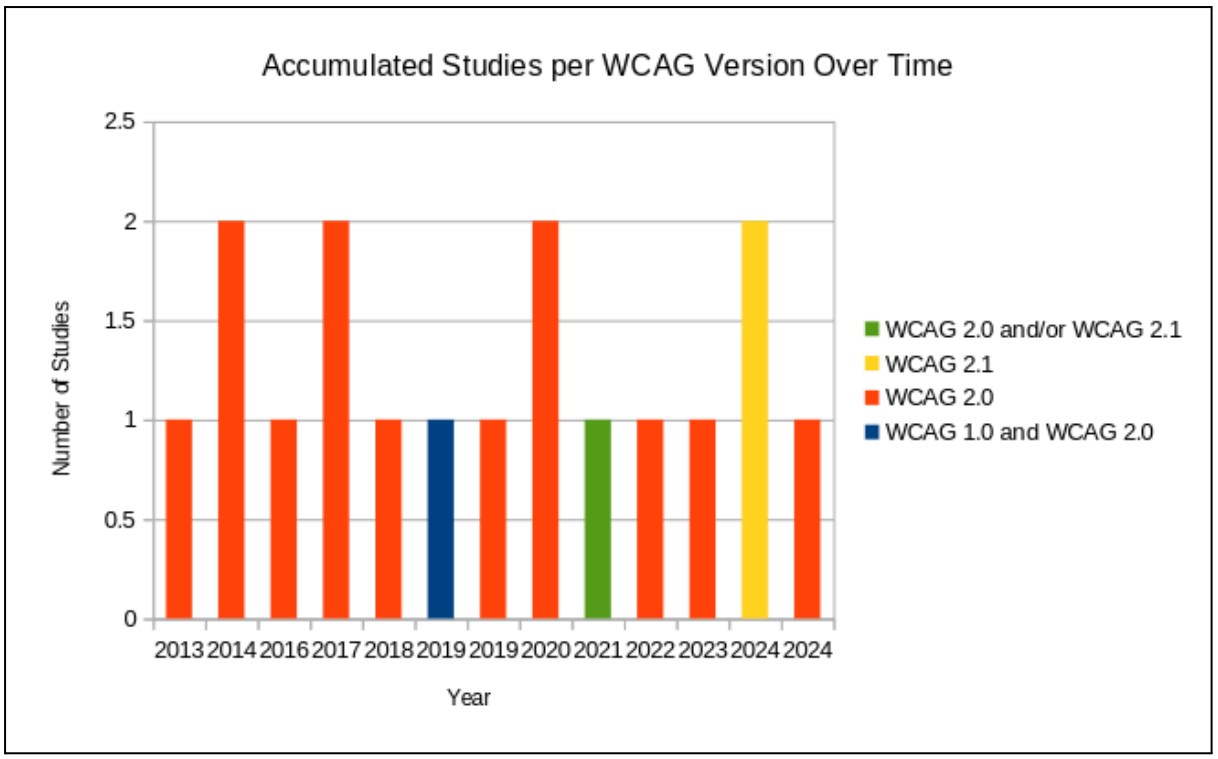
Notes were also taken separately in a text document, where initial remarks, findings and some notes were recorded. Finally, in a spreadsheet, each paper was recorded along with information and the themes for a better overview."

Reporting

Using the summarised data and the recorded themes it was possible to create the figures and write the review, where each study was numbered and referred to, along with findings and themes.

Table 1: List of the studies with summaries of findings.

Paper	Findings
[14]	"very few Indian e-government websites adhere to WCAG standards for accessibility"
[15]	"majority of the university websites in the study did not meet the WCAG 2.0 accessibility criteria"
[16]	"university hospital websites in Turkey had low compliance levels according to the WCAG 2.0 guidelines"
[7]	A proposed resource for testing accessibility tools
[8]	Introduced a new numerical metric
[9]	"Relying on just automated tests entails that 1 of 2 success criteria will not even be analysed and only 4 out of 10 in those analysed will be caught"
[2]	"This study highlights the strengths and weaknesses of selected automatic accessibility testing tools"
[10]	"we suggest different tools should be utilized to provide consistency and obtain reliable data from online evaluation tools, thereby improving tool effectiveness"
[5]	"The results obtained illustrate the interest of our accessibility indicator"
[17]	our study shows that the minimum level of WCAG 2.0 accessibility conformance level was not met by the tested Saudi e-government sites"
[18]	"findings reveal that there are several accessibility issues …"
[6]	"presents … analysis of … (WCAG) 2.0 + accompanying documents" [for] "… automatic checking tool" [+] "… definition of a web accessibility metric"
[12]	"presents an overview of a tool designed for web accessibility monitoring, auditing, and evaluating of government Qatari websites"
[13]	"proposes a combined approach, with the application of automatic and heuristic tools to make websites more accessible"
[11]	"demonstrate LLMs can augment automated accessibility testing to catch issues that pure software testing misses today"
[20]	"the websites under the study do not follow WCAG 2.0 principles and guidelines"
[19]	"none of the website satisfied completely the WCAG 2.0 guidelines"



Fig, 1. The different WCAG versions in the studies, as accumulated number of studies per WCAG version over time (year). The majority of the studies used WCAG version 2.0. One study was difficult to interpret whether it used WCAG 2.0 or WCAG "2.1" (green color).

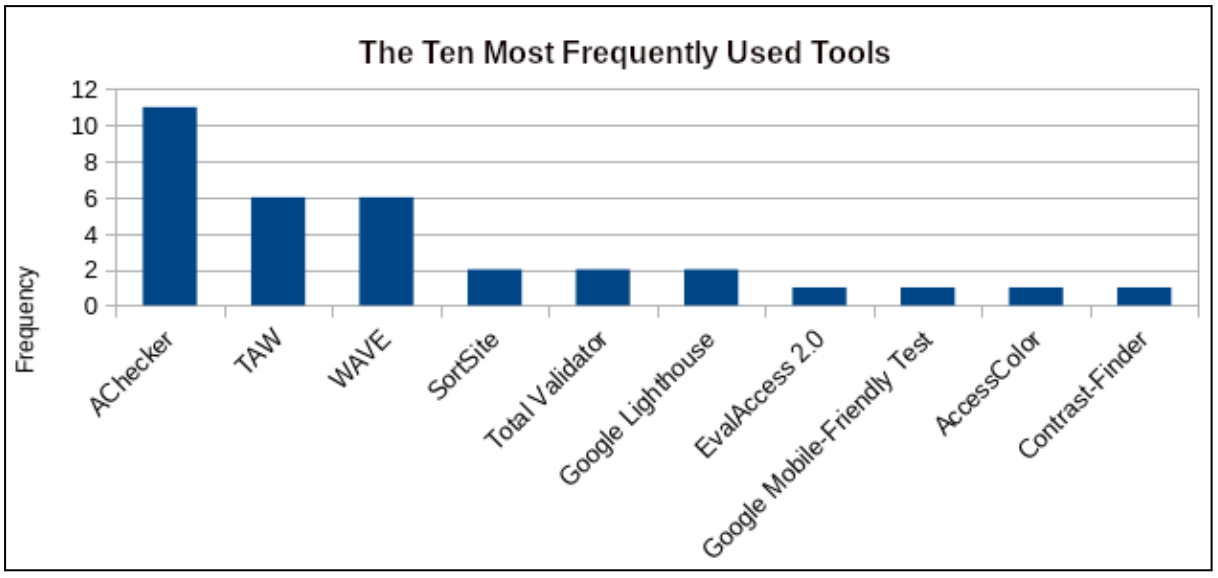


Fig. 2. The ten most frequently used tools in the studies.

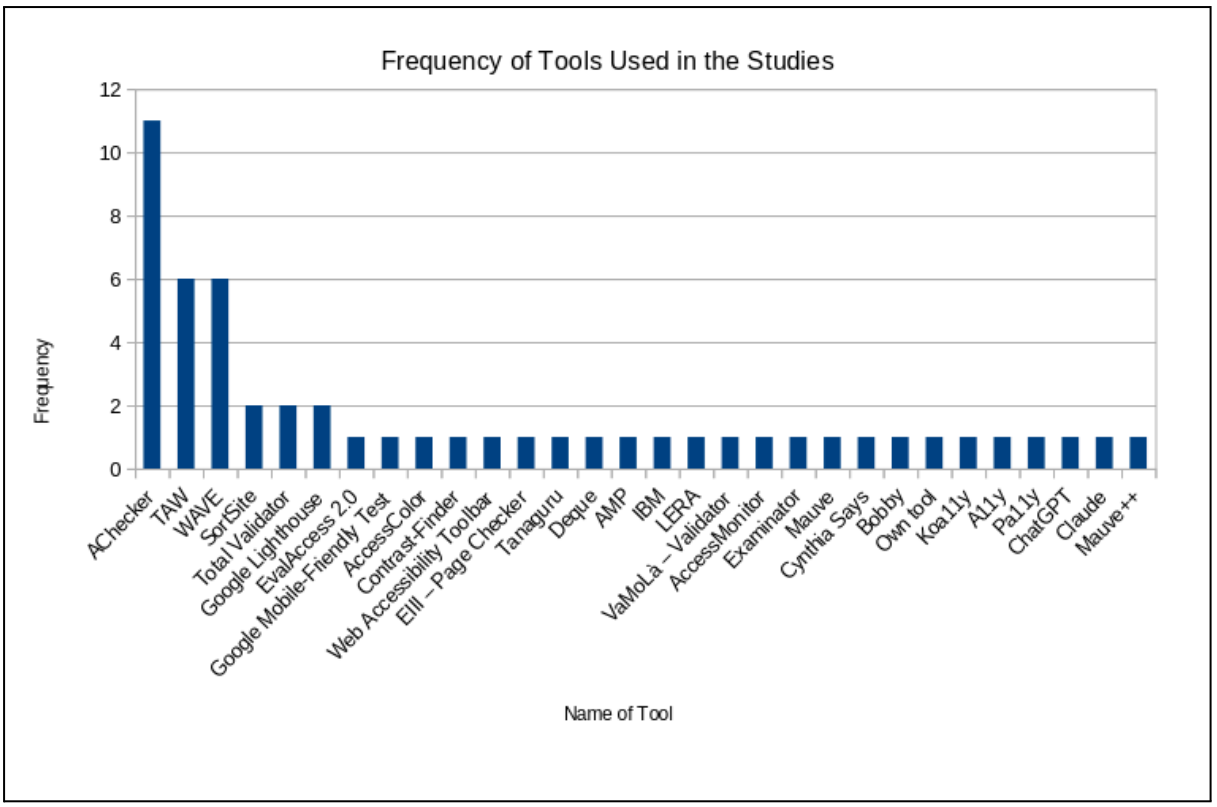


Fig. 3. The different tools used in the studies, may indicate commonly used tools.