

# Support for Accessible Software Coding: Results of a Rapid Literature Review

Cassio Andrade Furukawa, Michele dos Santos Soares,  
Maria Istela Cagnin and Débora Maria Barroso Paiva

Faculdade de Computação, Universidade Federal de Mato Grosso do Sul,  
Cidade Universitária, s/n - Campo Grande/MS, Brazil  
{*cassio.furukawa, michele.soares, istela.machado, debora.paiva*}@ufms.br

## Abstract

Due to difficulties faced by people with disabilities, it is necessary to provide all citizens with mechanisms to guarantee the accessibility of information and services offered through the internet. This study aims to identify which models and tools support developers in the software coding phase in order to meet accessibility requirements. The data obtained were extracted from the literature and were evaluated with professionals working with information technology. It was possible to conclude that taking accessibility into account throughout the software development cycle from the early stages, as well as having models and tools that help during the software coding process are essential to succeed in creating accessible products.

**Keywords:** accessibility, software engineering, coding phase, software development, guidelines

## 1 Introduction

The Covid-19 pandemic and, consequently, social distancing have made the need for access and knowledge even more evident and emerging so that the use of digital technological tools is possible [1]. Digital technology has become the center of relationships and practices in society and, therefore, access to resources provided through the internet should be guaranteed to all.

In this context, it is said that a website is accessible when it is understood, operated and perceived by users, regardless of their disabilities [2]. Therefore, the content of Web pages must be accessible as many people may not see, hear, have problems getting around or even have difficulties processing or understanding some type of information [3].

Although there is a need to provide accessibility in Web systems, some companies and software developers face difficulties in incorporating accessibility in their projects [4]. In order to help them and to ensure that systems are accessible so that people with disabilities can use them, international Web standards emerged, developed by the World Wide Web Consortium (W3C) [5].

These W3C standards aim to define the features that websites must contain to provide accessibility for people with disabilities. As an example, we can mention the WCAG guidelines (Web Content Accessibility Guidelines) [6]. Furthermore, in several countries there are laws that make it mandatory to adapt Web systems in order to be accessible to the population [7].

Still, not all available Web resources are accessible for people with disabilities to make use of. This may be related to some problems, such as (i) the lack of technical knowledge of engineers, analysts and software developers; (ii) the lack of tools that support the inclusion of accessibility throughout the software life cycle; and (iii) the prevailing software development culture of allocating insufficient resources (time and people) to the design and evaluation of graphical interfaces [8, 9].

In this sense, some works were developed with the aim of helping to solve these problems. An example is the work of Andrade et al. [10] who presented a method that integrates accessibility to the Software Engineering process and a tool called Acero that provides computational support for the proposed approach. The inclusion of accessibility is carried out in a transparent way, that is, when using the Acero tool, professionals will be able to generate accessible applications without the need to be experts in the area.

Regarding the support of the inclusion of accessibility throughout the software development cycle, the work developed by Oliveira et al. [11], culminated in the development of the framework called Homero, whose

objective is create HTML code that meets the accessibility rules proposed by WCAG 2.0. In this context, the development of accessible applications has become a social responsibility for the software engineering industry [12].

However, the lack of knowledge and adequate training for the technology community is one of the problems faced in creating accessible products [13]. It is important to emphasize that the subject is rarely addressed in the academic training of students and, in addition, there is no explicit information about accessibility and software development models used by the industry [14].

Thus, in order to identify which models and tools support developers in the software coding phase that meet accessibility rules and requirements, a Rapid Review (RR) of the Literature was carried out, aiming to provide evidence and important information that may be useful to information technology professionals with regard to the development of accessible software.

This work is organized as follows: Section 2 presents concepts about accessibility, tools and related works; Section 3 describes the research method used; Section 4 presents the results of the RR; Section 5 presents the evaluation of results based on the professionals' perception in the area of information technology through an opinion survey; Section 6 discusses the limitations and threats to the validity of this study. Finally, Section 7 concludes the work with final considerations and directions for future work.

## 2 Background

In this section, definitions and concepts useful to understand the contents used in this work are presented.

### 2.1 Accessibility and its Guidelines

The concept of accessibility arises from the need to include people with disabilities in society, in order to allow full participation and also contribute to improving the quality of life of these individuals [15]. When it comes to the Web, accessibility is the activity of making software applications more usable by as many people as possible, regardless of their different abilities [16].

In this way, accessibility can be understood as a process of adaptation of things and people (in relation to the way of thinking), so that this inclusion happens; and it can also be understood as a characteristic of everything that is accessible, that is, everything that has been suitable for use by people with disabilities [17].

Regarding the development of Web applications, accessibility is related to the suitability or adaptation of page codes so that they can work well with Assistive Technologies (ATs)<sup>1</sup>, as well as to support different types of interaction with interface components (e.g., keyboard and mouse navigation). Thus, in order to ensure that Web resources are accessible so that people with disabilities can use them, the main resources to be used are the accessibility guidelines created by the W3C.

The WCAG guideline refers to non-mandatory technical guidance and a set of documents that explains, through guidelines and recommendations, how to make Web content accessible so that people with disabilities can also make use of it [18]. The current version of WCAG is 2.1 and the document is structured around four principles (noticeable, operable, understandable and robust), twelve guidelines and three levels of compliance: A, AA and AAA (the first being the ranking of the least accessible sites, and the last being the most accessible sites, respectively).

For each principle, there is a set of guidelines that provide the basic goals and success criteria that developers must meet in order to make Web content more accessible to users with different types of disabilities. Following WCAG, software developers, applications and Web pages will be able to create content accessible to people with disabilities, including blindness and low vision, deafness and hearing loss, limited movement, speech impairment, light sensitivity and also guidelines for cognitive limitations and learning difficulties [18].

In addition to WCAG, there is another important standard also developed by the W3C, which is the WAI-ARIA (Web Accessibility Initiative - Accessible Rich Internet Applications), a technical specification that provides a framework that allows improving the accessibility of Web pages by marking the tag (marks that allow the creation of Web pages) of HTML [19].

Still in this scope, tools to detect accessibility problems play a fundamental role. Alonso et al. [20] suggest using automated tools to check websites for WCAG compliance more effectively. The next subsection presents an overview of these tools.

---

<sup>1</sup> Assistive Technologies are tools and devices that help to promote, maintain or improve the independence of people with disabilities [15].

## 2.2 Accessibility Tools

There is a wide range of tools with different purposes and formats in order to support the accessibility verification throughout the software development and also in its evaluation, in general such tools aim to detect accessibility problems. According to Alonso et al. [20], WCAG manual testability is unreliable, suggesting the need for tools that can more effectively verify WCAG compliance.

Some tools can discover a number of accessibility issues at the same time (e.g., Accessibility Checker<sup>2</sup>). There are also some automated tools that can be used to analyze the code and discover specific problems quickly, such as the a11y checker<sup>3</sup>, which aims to identify accessibility problems in HTML markup. Another very interesting tool worth mentioning is eslint-plugin-jsx-a11y<sup>4</sup>, it is a library (lib) that can be added to the project being developed and has the purpose of performing a static verification on the source code with the purpose of detecting accessibility issues in React apps.

In addition, the W3C has a page that contains a list of numerous tools that can be useful in the process of ensuring accessibility in Web applications. The tools can be selected according to different filters, such as: the tool's interface language, the accessibility standard used, the type of tool, whether it is a browser plugin, a website, a desktop program or a mobile application [21].

## 2.3 Related Works

Software Engineering plays an essential role in the process of developing accessible applications, as it is able to promote the integration between methodologies, techniques and specific accessibility activities throughout the software development [22]. With this, researchers have pointed out the importance of considering accessibility throughout the software life cycle [23, 24, 25].

Among the related works that aim to investigate accessibility throughout the software development cycle, there is the study by Paiva et al. [22], in which the authors carried out a systematic literature review (RSL) on accessibility and how it fits into software development processes. To achieve the objective, the authors investigated guidelines, techniques and methods presented in the literature and, from there, it was possible to provide designers and developers with methods that can support the process of enrichment and enhancement of accessibility. The study also presented some gaps and challenges that deserve to be investigated, among them, the lack of methods and tools to correct problems related to lack of accessibility in software. In this sense, our work aims to help mitigate this problem, highlighting studies that address this issue and present a model or tool that supports the development of accessible software.

In the study by Bi et al. [4], the authors sought to understand whether accessibility is considered in practice, what accessibility problems are faced by developers, what causes accessibility problems and what solutions are used to address accessibility problems. To achieve the goals, they collected 11,820 accessibility issues that developers discussed on popular GitHub projects. In doing so, the authors provide general recommendations for dealing with accessibility issues, as well as practical lessons on how to employ standards, tools, and APIs to ensure and facilitate the design and development of software that meets accessibility rules and requirements.

An empirical study aimed at understanding the accessibility of mobile applications developed for Android technology was conducted by Alshayban et al. [26]. The authors highlighted a large-scale analysis of the prevalence of a wide variety of accessibility issues in over 1,000 Android apps across 33 different app categories. With this, it was possible to identify 11 types of accessibility problems, for example, TextContrast and SpeakableText. The authors also released the results of a survey involving 66 professionals that reveal practices and challenges regarding accessibility in Android applications. Survey results show that developers are often unaware of accessibility principles and existing analysis tools are not sophisticated enough to use. This finding is very important because it is a motivating point for carrying out this research.

## 3 Research Method

RR is a kind of knowledge synthesis in which the elements of the systematic review process are simplified or omitted in order to generate information in a short period of time [27]. RR seeks to carry out a comprehensive and impartial research, based on a real prospected problem. In addition, RRs are secondary studies focused on providing evidence to practitioners in a timely manner at lower costs [28].

An RR has some key aspects [29]: (1) it reduces the costs of more bureaucratic methods; (2) provides evidence in a timely manner; (3) reports results through engaging media; and (4) works closely with profes-

<sup>2</sup><https://www.accessibilitychecker.org/>

<sup>3</sup><https://github.com/Muhnad/a11y-checker>

<sup>4</sup><https://github.com/jsx-eslint/eslint-plugin-jsx-a11y>

sionals. Finally, RRs follow systematic protocols, although some methodological decisions aimed at providing evidence in less time may be biased.

To perform the RR, we considered the guidelines provided by Cartaxo et al. [30], Tricco et al. [29] and Rico et al. [31]. In general, the main activities suggested by these authors are: (i) problem definition, (ii) specify the research questions, (iii) define the search string, (iv) search strategy, (v) data sources and inclusion and exclusion criteria, (vi) synthesis methods and (vii) initial proposals of how the results will be disseminated. The following subsections describe the study design and execution in detail.

### 3.1 Planning the Review

The planning and definition of the protocol that was used in this study had the involvement of professors and students of the research group in Software Engineering of the academic master's program in Computer Science at Universidade Federal de Mato Grosso do Sul (UFMS) and who have solid knowledge and experience in software engineering and accessibility.

The identified problems that motivated this study are related to the lack of training and knowledge on the part of developers and technology professionals in relation to accessibility, very abstract and distant models from coding and CASE tools that offer little help during the process of software development [32, 33, 10, 22].

Thus, the main objective of this study was to identify which models and tools support developers in the software coding phase that meet accessibility requirements. Based on this objective, the following research questions were defined:

**Research Question 1 (RQ1):** What are the main models or tools that support the developer in the coding phase in relation to producing accessible code?

**Research Question 2 (RQ2):** What are the main features of the existing tools that support the developer in the coding phase in relation to producing accessible code?

During the process of definition and construction of the search string, three studies were defined that were called “control articles”, as they are relevant to the scope of this study and consequently should be returned in the search using the search string. Thus, from these studies that are listed in Table 1, a set of keywords was selected (present in the title, abstract and keywords of the studies) that were combined with the research questions.

Table 1: Studies and keywords used to define the search string

| Title                                                                                     | Keywords                                                     | Reference |
|-------------------------------------------------------------------------------------------|--------------------------------------------------------------|-----------|
| Refining the Aceasy Framework from an accessibility assessment                            | accessibility, Web development                               | [34]      |
| Checking and correcting the source code of Web pages for accessibility                    | accessibility, checking, correcting, source code             | [35]      |
| AccTrace: Accessibility in Phases of Requirements Engineering, Design and Coding Software | accessibility, implementation, coding phase, coding software | [36]      |

The set of keywords was tested in different databases in order to test whether the control articles returned in the results and, consequently, the keywords could be improved and refined iteratively, as well as synonyms and variations of the keywords were established to find a set of relevant studies. Finally, using logical operators to connect the keywords, the search string looked like this:

*((“software implementation phase” OR “software coding phase” OR “coding software” OR “correcting source code” OR “checking source code” OR “implementation phase” OR “coding phase” OR “web development”) AND accessibility).*

After defining the search string, the used search strategy followed the following procedures: (i) search and extract the studies in the IEEE Xplore<sup>5</sup> and Scopus<sup>6</sup> databases using the search string; (ii) eliminate duplicate studies; (iii) filter studies by title, keywords and analysis of abstracts using inclusion and exclusion criteria; (iv) read the selected studies in full; (v) extract the data (using an extraction form); (vi) summarize the results and (vii) disseminate the results.

The use of only two databases was due to the fact that Scopus indexes and carries out research in several relevant digital libraries on Software Engineering [28], returning studies published in various journals and conference proceedings. Furthermore, considering that the RRs have a short period of execution, it was decided to use only the two databases mentioned for the search and obtainment of the studies.

<sup>5</sup><https://ieeexplore.ieee.org>

<sup>6</sup><https://www.scopus.com>

The selection procedure was based on inclusion and exclusion criteria. For the inclusion criteria, studies related to the theme of Web accessibility and that present a model or tool that supports accessibility during the software coding phase were considered.

The exclusion criteria (EC) used were the following:

- **EC1:** Studies that are not written in English or Portuguese;
- **EC2:** Studies that are not peer-reviewed (for example, preface, book, editorial, abstract, poster, panel, lecture, roundtable, workshop or demonstration) and work in progress;
- **EC3:** Studies that do not meet the inclusion criteria;
- **EC4:** Studies not fully accessible in electronic format.

Two researchers jointly developed the review protocol, which was reviewed and evaluated by the other researchers involved in this study and also by professors from the research group.

### 3.2 Conducting the Quick Review

One researcher (first author) performed activities 1, 2, 3 and 4, and the other researchers involved in the study were responsible for proceeding with the review and evaluation of these steps performed by the first author. The summary of the selection steps can be seen in Figure 1, and will be detailed below:

1. Using the search string, a researcher performed searches in the IEEE Xplore e Scopus databases, obtaining a total of 170 studies, 16 in IEEE Xplore and 154 in Scopus;
2. Subsequently, 10 duplicate studies were removed, leaving a total of 160 studies;
3. Applying the inclusion and exclusion criteria 152 studies were eliminated;
4. Finally, 8 studies remained for the complete reading and data extraction, which are listed in Table 2.

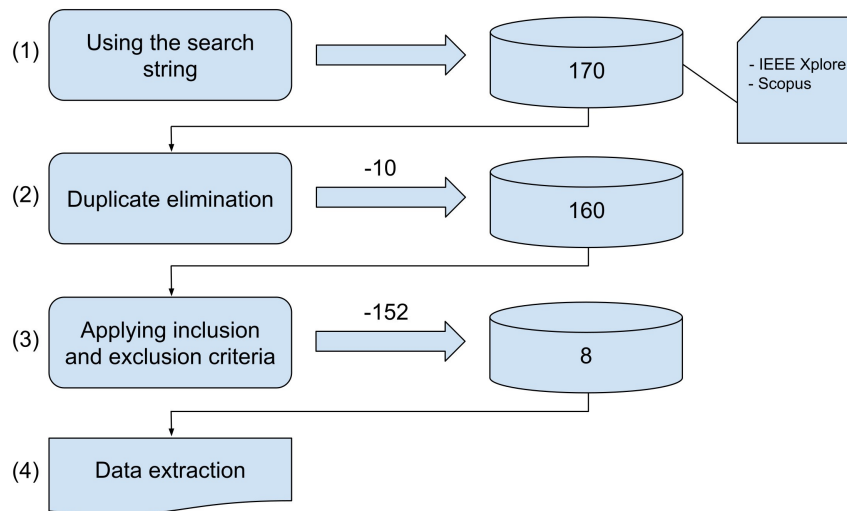


Figure 1: Summary of Selection Steps

The Table 3 presents a summary of the exclusion criteria, informing the number of studies eliminated in each criterion. An explanation is needed about the number of studies eliminated in the EC3 exclusion criterion, this is due to the fact that several studies eliminated in this criterion had evaluation tools and/or had no relation to the scope of this study. A spreadsheet containing information about the studies and inclusion and exclusion criteria that were applied is available at <https://bityli.com/ByCM0h>.

Table 2: Selected studies for full reading and data extraction

| ID | Title                                                                                                                                                    | Presented    | Year | Reference |
|----|----------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|------|-----------|
| 01 | Refining the Aceasy Framework from an accessibility assessment                                                                                           | Tool         | 2021 | [34]      |
| 02 | A model-driven approach to cross-platform development of accessible business apps                                                                        | Model        | 2020 | [37]      |
| 03 | AccTrace: Accessibility in Phases of Requirements Engineering, Design and Coding Software & accessibility, implementation, coding phase, coding software | Tool         | 2014 | [36]      |
| 04 | Integrating accessibility as a quality property in Web developments                                                                                      | Model        | 2013 | [25]      |
| 05 | A framework for building Web sites that are friendly to visually impaired                                                                                | Tool         | 2012 | [12]      |
| 06 | Checking and Correcting the Source Code of Web Pages for Accessibility                                                                                   | Model + Tool | 2012 | [35]      |
| 07 | Increasing the quality of Web systems: By inserting requirements of accessibility and usability                                                          | Model        | 2012 | [38]      |
| 08 | A Web design framework for improved accessibility for people with disabilities (WDFAD)                                                                   | Model        | 2008 | [39]      |

Table 3: Summary of Exclusion Criteria

| Criteria                                                                                                                                                                       | Amount |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| EC1. Studies that are not written in English or Portuguese                                                                                                                     | 6      |
| EC2. Studies that are not peer-reviewed (for example, preface, book, editorial, abstract, poster, panel, lecture, roundtable, workshop or demonstration) and works in progress | 5      |
| EC3. Studies that do not meet the inclusion criteria                                                                                                                           | 140    |
| EC4. Studies not fully accessible in electronic format                                                                                                                         | 1      |

### 3.3 Data Extraction

At this stage, relevant data that could help answer the research questions were extracted and a form was created and used to document, store and organize the data extracted from the studies. Table 4 shows the form that was used to document the information extracted from the studies.

Table 4: Information extracted from the form of the selected studies

| Information               | Options                       |
|---------------------------|-------------------------------|
| 1. Paper ID               |                               |
| 2. Title                  |                               |
| 3. Model or Tool          |                               |
| 4. Researchers Summary    |                               |
| 5. Context Description    | Yes(which?)/No                |
| 6. Presented Tool         | Yes(what characteristics?)/No |
| 7. Evaluation Description | Yes(which?)/No                |
| 8. Year                   |                               |

## 4 Results

The RR was carried out in the period from September 15, 2021 to March 11, 2022, performing the phases mentioned in the previous subsections.

As described in subsection 3.2, 8 studies were selected to answer the research questions, of which 3 studies present a tool, 4 studies present a model and 1 study presents a model and tool. The results are described below.

**RQ1:** What are the main models or tools that support the developer in the coding phase in relation to producing accessible code?

When analyzing the selected studies, models and tools were found that are intended to support developers during the software development process in order to meet accessibility requirements, more precisely in the coding phase. Next, the models and tools found in the selected studies will be presented.

### 4.1 Models

The study by Vu et al. [35] presents a model to verify and correct the source code of Web pages, taking into account accessibility and its corresponding structure. In the proposed processing model, inputs are defined by an HTML page on the client side and a page with source code on the server side, which is processed through four steps in order to achieve compliance with WCAG 2.0.

According to the authors, both inputs go through parsers so that it is possible to obtain the client and server nodes represented by tags. Next, the client nodes are checked for compliance with WCAG according

to techniques contained in WCAG. In the next (third) step, efforts are focused on discovering server nodes that have violations of WCAG recommendations. This is done by mapping only non-compliant client nodes to server nodes. The fourth step is performed semi-automatically and focuses on fixing noncompliant server nodes.

In order to evaluate the proposed model, the authors implemented a tool called CHECORSER, which has the function of checking and correcting ASP language scripts according to WCAG rules. The authors mention that they tested the use of the tool, which worked efficiently for the MVC (Model, View, Controller) model, however, they did not present details of this evaluation.

A Web Design Framework to Improve Accessibility for People with Disabilities (WDFAD), which provides a set of guidelines on how to develop accessible Web applications for people with visual impairment, was presented by Baguma and Lubega [39]. WDFAD provides a set of guidelines for developers to be able to develop accessible Web applications. These guidelines are based on the three elements of Web applications, namely: content, navigation, and user interface. Using the Non Functional Requirements (NFR) framework, Web accessibility design goals are represented as primary and secondary goals.

Primary goals represent high-level Web accessibility design goals, such as: content accessibility (A), navigation accessibility (B), and user interface accessibility (C). On the other hand, secondary objectives represent what needs to be done (Web accessibility requirements) to achieve the primary objectives. WDFAD aims to make Web accessibility guidelines easier for developers to understand and apply, so that it is possible to create accessible resources for people with disabilities, especially those with visual impairments [39].

A model whose solution is user-centered, iterative, accessibility-aware and evolutionary, which values accessibility from the beginning of the project is proposed by Reichling and Cherfi [25]. In the approach proposed by the authors, an iterative process containing three phases is presented: analysis, design and evaluation.

The analysis phase aims to understand the specific requirements and needs and define the context of use and also contains a specification of goals and objectives, containing accessibility requirements. The design phase is about how to produce a design that meets the requirements of the users. The evaluation phase intends to evaluate the quality of the product and this means identifying eventual accessibility problems or evaluating if the requirements are met, if it complies with the accessibility guidelines and if it satisfies the expectations of the end users, obtaining feedback [25]. In addition, by proposing that the phases are iterative and evolutionary, agile methodologies are combined with the integration of accessibility criteria in Web development.

In Dias et al. [38] a model is proposed in order to include accessibility requirements in a Web engineering framework. For this, the authors extended a classic model to the process of developing Web applications aiming to include accessibility and usability in non functional requirements during development from the early stages. Based on this, the main result obtained was a list of non-functional requirements in compliance with WCAG accessibility guidelines and usability heuristics formalized by Jakob Nielsen and Rolf Molich, which must be considered by Web engineering during the development of Web systems.

In Rieger et al. [37] a model-driven framework is presented, which aims to develop model-driven mobile applications, based on the framework Xtext<sup>7</sup>, which allows the creation of textual languages, provides a Domain Specific Language (DSL) for specifying user interfaces, data model, and workflows of data manipulation steps for the entire application. The original structure of the model-driven framework did not have dedicated support for accessibility features, so the authors extended the model-driven framework considering accessibility requirements so that the generated applications would support those requirements and this proposed framework would be compliant with accessibility guidelines.

## 4.2 Tools

The study by Germano et al. [34] presented a front-end framework for creating responsive Web applications with accessibility for the visually impaired called Aceasy, based on guidelines WCAG 2.0. The purpose of this framework is to facilitate the creation of Web pages by developers through the generation of HTML code. The framework that uses the JavaScript programming language implements some functions whose intention is to return a list of objects that represent the contents that will be translated into tags and will constitute the page that will be built in the HTML text markup language.

To evaluate the Aceasy framework, a website containing basic mathematical operations (addition, subtraction, multiplication and division) was developed using the framework. The results show that the functions present in Aceasy generate accessible HTML code, being able to improve the interaction of the visually

<sup>7</sup>Xtext, which is an open source framework for developing programming languages and domain-specific languages (DSLs). Unlike standard parser generators, Xtext not only generates a parser, but also a class template for the Abstract Syntax Tree and a fully customizable Eclipse-based Integrated Development Environment (IDE) with all resources [40].

impaired with websites. According to the authors, the results indicate that the framework lacks the implementation of some elements so that it is able to generate skip links<sup>8</sup> and a manager of headers and their hierarchies.

A tool called AccTrace (Eclipse plugin) was developed by Branco et al. [36] in order to allow traceability of accessibility requirements and provide relevant information for the developer in building accessible products. Making use of the tool, it is possible to evolve the accessibility requirements up to the software coding phase. In general, comments are added to the source code (more precisely in the class scope) indicating what should be included when writing the source code to meet accessibility requirements. The tool was evaluated through a proof of concept, where a software project (Web search engine) was created, the evaluation results according to the authors indicate that the objectives of the proposed approach were achieved.

A tool called CHECORSER for checking and correcting Web pages that make use of ASP technology was presented by Vu et al. [35]. The tool allows developers to verify and correct server source pages, rather than just checking WCAG compliance for HTML client pages. The tool is very useful as it is able to identify possible violations of accessibility rules, in order to allow developers to make the necessary corrections during software development. However, the authors did not describe the evaluation of the tool in question.

Wanniarachchi and Jayathilake [12] propose a framework in the form of a tool called “Web Accessibility Generator”, whose purpose is to automate the process of developing accessible websites. The Web Accessibility Generator is intended to be a comprehensive framework that can be used by Web developers during the process of creating or modifying websites for accessibility. Therefore, the proposed solution suggests an interactive mechanism with the Web developer to detect accessibility problems and correct them. The accessibility rules defined in the Web Accessibility Generator target users with visual impairments, and finally, the corrections of problems related to detected accessibility violations must be done manually by the Web developers [12].

### 4.3 Main features of the tools presented in RQ1

Based on studies that presented some tools, the main characteristics of the tools mentioned in the answer to **RQ1** are presented in Table 5 and aim to answer **RQ2**.

**RQ2:** What are the main features of the existing tools that support the developer in the coding phase in relation to producing accessible code?

From the data extracted from the selected studies, we can observe that all of them present and follow some accessibility guidelines. It is worth noting that WCAG was cited in all studies, version 2.0 of the aforementioned standard was followed by 6 studies, version 2.1 by 1 study, one study did not report the version of WCAG it was following and this study also followed eMAG (Accessibility Model in Electronic Government)<sup>910</sup>.

With this, one can conclude the importance of accessibility guidelines to support the creation of accessible products. In addition, the guidelines are used to guide the creation of models and tools that help developers in the process of developing software that meets accessibility requirements.

## 5 Evaluation of Results and Discussion

As the purpose of the RRs is to provide information and evidence to professionals in a timely manner and thus contribute to solving practical problems faced by them, the evaluation of data extracted from the literature with professionals in the technology area is necessary.

Thus, we conducted an opinion survey with professionals in the technology area, in which they were able to answer a form containing questions that were prepared based on data extracted from the literature. The aim was to obtain the opinion of these professionals based on their perceptions of the data extracted from the literature, identifying whether or not they agreed with the findings based on the selected studies and whether such findings are useful and relevant to be able to solve the prospected problem that motivated this RR.

The form was available to be answered from 08:00 am on February 1, 2022 until 11:59 pm on February 7, 2022. During this period, 20 responses were submitted and it is estimated that the form was delivered to approximately 200 professionals (the exact number could not be determined as the professionals shared the form with some of their contacts).

<sup>8</sup>Skip Link is an internal link at the beginning of a hypertext document that allows users to skip browsing material and quickly access the main content of the document [41].

<sup>9</sup>The eMag is a document that contains a set of recommendations with the purpose of allowing and assisting in the implementation of digital accessibility on websites of Brazilian government institutions, its elaboration was based on the guideline WCAG [42].

<sup>10</sup><https://emag.governoeletronico.gov.br>

Table 5: Main features of the tools mentioned in RQ1

| Tool                        | Features                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Reference |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| Acceasy                     | Enables the creation of HTML5 codes (language used to create of the pages of the sites, Hypertext Markup Language) indirectly, through the use of JavaScript (Web programming language), with this, from the framework it is possible to create the elements (tags) that make up a Web page and these already meet the rules of accessibility. In addition, the generated code is responsive, that is, it works properly on devices with different screen sizes, so that on each of them the interface components are automatically adjusted, providing a suitable user experience for each user device. | [34]      |
| AccTrace                    | Enables traceability of accessibility requirements, generates feedback in the source code (more precisely in the class scope), based on the information described in the requirements, which have information about accessibility. It is worth noting that this tool does not perform source code verification, but provides important information in the source code that helps the developer during the development process.                                                                                                                                                                           | [36]      |
| CHECORSER                   | Allows developers to (i) automatically identify and generate the missing elements (including nodes, attributes) of an ASP.NET page; (ii) enter the default values of these missing elements for this page; (iii) identify violations in the source code; (iv) receive suggestions to correct violations on server origin pages; and (v) perform automatic code correction of the server's source pages.                                                                                                                                                                                                  | [35]      |
| Web Accessibility Generator | Allows you to scan the Web application code and identify potential improvements accessibility concerns when developing a new or modifying an existing site. The tool contains four main visualization objects, namely: (i) source code editor, which highlights elements with violations and allows the user to make changes; (ii) results panel, which displays the list of accessibility violations; (iii) panel that displays result details and (iv) progress and summary bar, which displays overall progress and a summary of accessibility violations remediation.                                | [12]      |

One of the questions sought to identify whether the professionals had knowledge about Web accessibility, 12 participants said they did not have knowledge about Web accessibility, which represents most of the answers, and this is in accordance with what was reported by Sirikitsathian et al. [32] who found a lack of knowledge on the part of IT professionals about accessibility, thus, the responses of these participants were discarded. Thus, after carrying out this screening, we were left with a total of 8 responses, this is our sample, which for this study only takes into account the responses of participants who said they had knowledge of Web accessibility.

All the information below is related to the professionals who answered the form and who make up the selected sample. Based on this, the academic profile of the professionals was as follows: 25% Undergraduate, 25% Specialization, 37.5% Master's and 12.5% Post-Doctorate. Where it is possible to observe that the highest percentage is represented by participants who have a master's degree. This can be explained by the fact that several contacts of the authors in the industry have already passed through the academy, in addition, professionals who have a background in research tend to be more receptive to responding to academic research.

Regarding the level of knowledge in programming or development of Web systems: 62,5% said they had advanced knowledge, 12.5% intermediate and 25% basic. Among the knowledge in accessibility reported by the participants, the following stand out: (1) knowledge about WAI-ARIA attributes for images, labels and forms; (2) knowledge about eMAG (Electronic Government Accessibility Model); (3) Accessibility on government portals; (4) accessibility toolbar items; and (5) use of settings and attributes that facilitate reading for the visually impaired.

A total of 100% of professionals agreed that a possible solution to make it possible to build products accessible to people with some type of disability would be to consider accessibility during the entire software development cycle from the initial stages, accordance with what was reported by Paiva et al. [22]. Furthermore, when it comes to software development, all professionals agree that using tools that support and assist during the development process (coding) can be useful in creating systems accessible to people with some type of disability.

Regarding the importance of having models (diagrams containing process flow, guides) that can be followed and tools that help during the software development process, 100% of professionals responded that models and tools can be useful to obtain success in creating systems that meet accessibility rules

and requirements, something that was already expected by the authors of this study, but which could be confirmed by the professionals who responded to the opinion survey.

Through an open question provided at the end of the questionnaire, it was possible to obtain feedback from technology professionals who participated in the opinion survey, on what they expect software coding tools to offer in terms of features that help create more accessible products. Some suggestions from professionals were:

- Conduct a source code check of the software being developed for violations or accessibility issues and send alerts warning of elements of the source code that do not comply with the guidelines, in addition, present implementation examples that help to solve the identified problem;
- Display a list of accessibility rules with implementation examples;
- Provide a specialized accessibility linter<sup>11</sup> to perform source code verification for accessibility violations;
- Facilitate the automatic generation of source code so that standard attributes are included to make the code accessible, for example: when inserting an `<img>` tag for images, also insert the attribute that makes the image accessible (attribute “alt” for the tag `<img>`) with the default value, so the code would look like `<img alt = “”>` conforming to WCAG 2.1 Level A Criterion 1.1.1.

The results of the opinion survey with the professionals were considered satisfactory by the authors, because, in addition to being able to evaluate the data extracted from the literature from the perspective of the perception of IT professionals, it also served to identify what they expect from the tools to offer in terms of characteristics, this feedback can guide future work.

It was also possible to identify that several professionals do not have knowledge about Web accessibility, something that confirms what is reported in other academic works such as those by Bi et al. [4] and Sirikitsathian et al. [32]. In addition, it was possible to confirm that accessibility issues being incorporated throughout the software development cycle from the early stages increases the chances of being able to build software accessible for people with disabilities.

## 6 Limitations and Threats to Validity

Threats to internal validity refers to aspects that can somehow affect the results obtained in the study [27], as can be seen below: (1) two databases were used to search for articles in order to increase the number of studies found, but at the same time using few databases can restrict the number of studies found; (2) the entire selection process was conducted by a solo researcher and reviewed by the other researchers involved in the study in order to validate what is included in the protocol and also the findings extracted from the selected studies; (3) an important limitation with regard to the evaluation of results with professionals in the technology area is that the perceptions of the participants who responded to the form were evaluated based on data extracted from the literature, that is, the perceptions could be different if other studies were selected from other search bases.

Another point to be mentioned in the threats to validity is the fact that professionals participated in the study only in the phase of evaluation of data extracted from the literature and not from the conception of planning and definition of the research protocol as recommended by Cartaxo et al. [28]. However, the results of this study show that this did not harm or negatively impact the performance of this research, as the results of this study are in accordance with what has been reported in works related to ours, as an example we can mention the works of Paiva et al. [22] and Bi et al. [4].

## 7 Final considerations and future works

This work sought to identify which models and tools can support developers in the software coding phase, so that it is possible to meet accessibility requirements. The data obtained were extracted from the literature and were evaluated together with professionals in the area of information technology through an opinion survey. With this, it was possible to conclude that taking into account accessibility throughout the software development cycle from the early stages, as well as having models and tools that help developers during the software coding process, are essential for success in creating accessible products.

Regarding future work, considering the results obtained in this RR, it is intended to develop a conceptual model focused on source code static verification and to develop a lint code verification tool that can help

<sup>11</sup>Linter is a tool to identify and report patterns found in code, in order to make the code more consistent and avoid bugs, in this way, linter statically analyzes the code in order to find problems quickly [43].

software developers to create Web systems that meet accessibility rules and requirements. In addition, it is expected that the tool will be able to verify the source code being developed, alert developers in case of violations of accessibility rules according to WCAG guidelines and propose tips for such accessibility violations to be resolved, in this way, to contribute to the creation of accessible products. In addition, it may be interesting to complement this work with snowballing.

## References

- [1] M. Malave, “The role of social media during the pandemic,” *Instituto Fiocruz – IFF*, pp. 453–469, 2020. [Online]. Available: <http://157.86.6.63/index.php/component/content/article/8-noticias/675-papel-redes-sociais>
- [2] R. Pezzotti, M. Sousa, and L. Martínez, “Guidelines for developing social networking mobile apps to deaf audience: a proposal based on user experience and technical issues,” in *IHC 2018 Proceedings of the 17th Brazilian Symposium on Human Factors in Computing Systems. ACM Digital Library, Belém, Brazil*, no. 26, 2018, p. 1. [Online]. Available: <https://doi.org/10.1145/3274192.3274218>
- [3] S. Sanchez-Gordon, M. Sánchez-Gordón, M. Yilmaz, and R. V. O’Connor, “Integration of accessibility design patterns with the software implementation process of iso/iec 29110,” *Journal of Software: Evolution and Process, Wiley Online Library*, vol. 31, no. 1, p. e1987, 2019. [Online]. Available: <https://doi.org/10.1002/smr.1987>
- [4] T. Bi, X. Xia, D. Lo, and A. Aleti, “A first look at accessibility issues in popular github projects,” in *2021 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 2021, pp. 390–401. [Online]. Available: <https://doi.org/10.1109/ICSME52107.2021.00041>
- [5] W3C, “About W3C,” <https://www.w3.org/Consortium/>, 2022, Accessed: 04/06/2022.
- [6] —, “W3C Accessibility Standards Overview,” <https://www.w3.org/WAI/standards-guidelines/>, 2022, Accessed: 04/06/2022.
- [7] —, “Web Accessibility Laws and Policies,” <https://www.w3.org/WAI/policies/>, 2018, Accessed: 04/06/2022.
- [8] H. B. Paulsen, “Finding an optimal method for conducting accessibility evaluations of the norwegian tax administration website,” Master’s thesis, Oslo and Akershus University College of Applied Sciences, 2016. [Online]. Available: <https://oda.oslomet.no/oda-xmlui/handle/10642/3345>
- [9] P. P. Rau, L. Zhou, N. Sun, and R. Zhong, “Evaluation of web accessibility in china: changes from 2009 to 2013,” *Universal Access in the Information Society*, vol. 15, no. 2, pp. 297–303, 2016. [Online]. Available: <https://doi.org/10.1007/s10209-014-0385-9>
- [10] W. T. Andrade, R. G. d. Branco, M. I. Cagnin, and D. M. B. Paiva, “Incorporating accessibility elements to the software engineering process,” *Advances in Human-Computer Interaction, Hindawi*, vol. 2018, 2018. [Online]. Available: <https://doi.org/10.1155/2018/1389208>
- [11] R. C. d. Oliveira, A. P. Freire, D. M. B. Paiva, M. I. Cagnin, and H. Rubinsztein, “A framework to facilitate the implementation of technical aspects of web accessibility,” in *Proceedings of the 16th International Conference on Universal Access in Human-Computer Interaction*. Springer, 2014, pp. 3–13. [Online]. Available: [https://doi.org/10.1007/978-3-319-07509-9\\_1](https://doi.org/10.1007/978-3-319-07509-9_1)
- [12] H. Wanniarachchi and D. Jayathilake, “A framework for building web sites that are friendly to visually impaired,” in *International Conference on Advances in ICT for Emerging Regions (ICTer2012)*. IEEE, 2012, pp. 103–110. [Online]. Available: <https://doi.org/10.1109/ICTer.2012.6422840>
- [13] L. Moreno, F. Valverde, P. Martínez, and O. Pastor, “Supporting navigation accessibility requirements in web engineering methods,” in *Journal of Web Engineering*. Rinton Press, Incorporated, 2013, p. 181–202. [Online]. Available: <https://dl.acm.org/doi/10.5555/2535629.2535630>
- [14] J. d. S. Silva, R. Gonçalves, and A. Pereira, “Accessibility in the software life cycle a maieutic exercise in software engineering,” in *2017 12th Iberian Conference on Information Systems and Technologies (CISTI)*. IEEE, 2017, pp. 1–5. [Online]. Available: <https://doi.org/10.23919/CISTI.2017.7975710>

- [15] R. S. Germano and M. A. Eliseo, “Acceasy-a front-end framework prototype for developing responsive web applications with accessibility for visually impaired people,” in *2020 15th Iberian Conference on Information Systems and Technologies (CISTI)*. IEEE, 2020, pp. 1–6. [Online]. Available: <https://doi.org/10.23919/CISTI49556.2020.9141063>
- [16] D. Hoffman, E. Grivel, and L. Battle, “Designing software architectures to facilitate accessible web applications,” *IBM Systems Journal*, vol. 44, no. 3, pp. 467–483, 2005. [Online]. Available: <https://doi.org/10.1147/sj.443.0467>
- [17] ONU, “Organização das nações unidas - accessibility and development—mainstreaming disability in the post-2015 development agenda,” 2015. [Online]. Available: [https://www.un.org/disabilities/documents/accessibility\\_and\\_development.pdf](https://www.un.org/disabilities/documents/accessibility_and_development.pdf)
- [18] W3C, “Web content accessibility guidelines 2.1.” <https://www.w3.org/TR/WCAG21/>, 2018, Accessed: 28/02/2022.
- [19] —, “Accessible Rich Internet Applications (WAI-ARIA) 1.1,” <https://www.w3.org/TR/wai-aria/>, 2017, Accessed: 04/06/2022.
- [20] F. Alonso, J. L. Fuertes, Á. L. González, and L. Martínez, “On the testability of wcag 2.0 for beginners,” in *Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A)*, 2010, pp. 1–9. [Online]. Available: <https://doi.org/10.1145/1805986.1806000>
- [21] W3C, “Web accessibility evaluation tools list,” <https://www.w3.org/WAI/ER/tools/>, 2022c, Accessed: 22/02/2022.
- [22] D. M. B. Paiva, A. P. Freire, and R. P. de Mattos Fortes, “Accessibility and software engineering processes: A systematic literature review,” vol. 171. Elsevier, 2021, p. 110819. [Online]. Available: <https://doi.org/10.1016/j.jss.2020.110819>
- [23] E. Krainz, K. Miesenberger, and J. Feiner, “Can we improve app accessibility with advanced development methods?” in *International Conference on Computers Helping People with Special Needs*. Springer, 2018, pp. 64–70. [Online]. Available: [https://doi.org/10.1007/978-3-319-94277-3\\_12](https://doi.org/10.1007/978-3-319-94277-3_12)
- [24] S. Horton and D. Sloan, “Accessibility in practice: a process-driven approach to accessibility,” in *Inclusive Designing*. Springer, 2014, pp. 105–115. [Online]. Available: [https://doi.org/10.1007/978-3-319-05095-9\\_10](https://doi.org/10.1007/978-3-319-05095-9_10)
- [25] M. Reichling and S. S.-s. Cherfi, “Integrating accessibility as a quality property in web developments,” in *IEEE 7th International Conference on Research Challenges in Information Science (RCIS)*. IEEE, 2013, pp. 1–12. [Online]. Available: <https://doi.org/10.1109/RCIS.2013.6577698>
- [26] A. Alshayban, I. Ahmed, and S. Malek, “Accessibility issues in android apps: state of affairs, sentiments, and ways forward,” in *42nd International Conference on Software Engineering (ICSE)*. IEEE/ACM, 2020, pp. 1323–1334. [Online]. Available: <https://doi.org/10.1145/3377811.3380392>
- [27] F. Ponce, G. Márquez, and H. Astudillo, “Migrating from monolithic architecture to microservices: A rapid review,” in *38th International Conference of the Chilean Computer Science Society (SCCC)*. IEEE, 2019, pp. 1–7. [Online]. Available: <https://doi.org/10.1109/SCCC49216.2019.8966423>
- [28] B. Cartaxo, G. Pinto, and S. Soares, “The role of rapid reviews in supporting decision-making in software engineering practice,” in *Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018*, 2018, pp. 24–34. [Online]. Available: <https://doi.org/10.1145/3210459.3210462>
- [29] A. C. Tricco, J. Antony, W. Zarin, L. Striffler, M. Ghassemi, J. Ivory, L. Perrier, B. Hutton, D. Moher, and S. E. Straus, “A scoping review of rapid review methods,” in *BMC medicine*, vol. 13, no. 1. BioMed Central, 2015, pp. 1–15. [Online]. Available: <https://doi.org/10.1186/s12916-015-0465-6>
- [30] B. Cartaxo, G. Pinto, and S. Soares, “Rapid reviews in software engineering,” in *Contemporary Empirical Methods in Software Engineering*. Springer, 2020, pp. 357–384. [Online]. Available: [https://doi.org/10.1007/978-3-030-32489-6\\_13](https://doi.org/10.1007/978-3-030-32489-6_13)
- [31] S. Rico, N. Ali, E. Engström, and M. Höst, “Guidelines for conducting interactive rapid reviews in software engineering - from a focus on technology transfer to knowledge exchange,” Zenodo, 2020. [Online]. Available: <https://doi.org/10.5281/zenodo.4327725>

- [32] P. Sirikitsathian, S. Chaveesuk, and C. Sathitwiriawong, “A conceptual framework for better understanding of factors influencing accessibility to a website and its acceptance by university students with visual impairments,” in *9th International Conference on Information Technology and Electrical Engineering (ICITEE)*. IEEE, 2017, pp. 1–5. [Online]. Available: <https://doi.org/10.1109/ICITEED.2017.8250493>
- [33] P. T. A. Junior and M. Piazza, “Framework interface components for accessibility issues in e-commerce,” in *Advances in Ergonomics Modeling, Usability & Special Populations*. Springer, 2017, vol. 486, pp. 527–539. [Online]. Available: [https://doi.org/10.1007/978-3-319-41685-4\\_47](https://doi.org/10.1007/978-3-319-41685-4_47)
- [34] R. S. Germano, T. A. N. de Sousa, and M. A. Eliseo, “Refining the acceasy framework from an accessibility assessment,” in *16th Iberian Conference on Information Systems and Technologies (CISTI)*. IEEE, 2021, pp. 1–6. [Online]. Available: <https://doi.org/10.23919/CISTI52073.2021.9476366>
- [35] T.-H.-G. Vu, D. T. Tuan, and V.-H. Phan, “Checking and correcting the source code of web pages for accessibility,” in *IEEE RIVF International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future*. IEEE, 2012, pp. 1–4. [Online]. Available: <https://doi.org/10.1109/rivf.2012.6169872>
- [36] R. G. Branco, M. I. Cagnin, and D. M. B. Paiva, “Acctrace: accessibility in phases of requirements engineering, design, and coding software,” in *2014 14th International Conference on Computational Science and Its Applications*. IEEE, 2014, pp. 225–228. [Online]. Available: <https://doi.org/10.1109/ICCSA.2014.51>
- [37] C. Rieger, D. Lucrédio, R. P. M. Fortes, H. Kuchen, F. Dias, and L. Duarte, “A model-driven approach to cross-platform development of accessible business apps,” in *Proceedings of the 35th Annual ACM Symposium on Applied Computing*. ACM, 2020, pp. 984–993. [Online]. Available: <https://doi.org/10.1145/3341105.3375765>
- [38] A. L. Dias, R. P. de Mattos Fortes, and P. C. Masiero, “Increasing the quality of web systems: By inserting requirements of accessibility and usability,” in *Eighth International Conference on the Quality of Information and Communications Technology*. IEEE, 2012, pp. 224–229. [Online]. Available: <https://doi.org/10.1109/QUATIC.2012.33>
- [39] R. Baguma and J. T. Lubega, “A web design framework for improved accessibility for people with disabilities (wdfad),” in *Proceedings of the 2008 international cross-disciplinary conference on Web accessibility (W4A)*. ACM, 2008, pp. 134–140. [Online]. Available: <https://doi.org/10.1145/1368044.1368077>
- [40] Xtext, “Language engineering for everyone!” <https://www.eclipse.org/Xtext/>, 2015, Accessed: 12/02/2022.
- [41] WebAIM, “Skip Navigation Links,” <https://webaim.org/techniques/skipnav/>, 2021, Accessed: 10/07/2022.
- [42] eMAG, “eMAG - Accessibility Model in Electronic Government,” <https://emag.governoeletronico.gov.br/>, 2014, Accessed: 12/06/2022.
- [43] ESLint, “Eslint - find and fix problems in your javascript code,” <https://eslint.org/>, 2022, Accessed: 16/02/2022.