A STUDY ON DISCOVERING ACCESSIBILITY ISSUES IN THE SOFTWARE DEVELOPMENT PROCESS

Luciano Arruda Teran - UNIVERSIDADE FEDERAL DO PARÁ - Orcid: https://orcid.org/0000-0002-0023-4179

Alan Trindade De Almeida Silva - UNIVERSIDADE FEDERAL DO PARÁ - Orcid: https://orcid.org/0000-0001-9325-0209

Giselle Lorrane Nobre Melo - UNIVERSIDADE FEDERAL DO PARÁ (UFPA) - Orcid: https://orcid.org/0000-0002-5285-1697

Marcelle Pereira Mota - UFPA - Orcid: https://orcid.org/0000-0001-9226-9020

Accessibility in computing, when well applied, allows for the inclusion of users and the breaking of barriers in the interaction process. However, approaches to support accessibility validation are commonly unknown or ignored. The goal of this study is to understand accessibility issues considered in the software development process. This understanding is the basis for overcoming existing challenges. We want to find out why software developers disregard accessibility issues to research how to minimize this problem. We can find research papers addressing specific accessibility issues, such as mobile or web systems, but not broadly covering these issues. This research has carried out a systematic literature mapping, a questionnaire with people involved in the software development cycle, and a mapping of tools for automated accessibility tests. These three ways of collecting data help get information about accessibility issues in different but still related contexts. The results indicate that although there are approaches to support developers, the tools are still limited regarding the type of platform, type of disability considered, or accessibility guidelines applied. In addition, tools are often focused on evaluation after software development, which makes the adjustment process more costly. The study’s theoretical contribution is that accessibility rarely appears as a requirement in the early stages of software development. Despite recognizing its importance, according to the data collected, accessibility supposedly should be an intrinsic feature of the software, sometimes ignored by the development team for a set of reasons. Finally, the social contribution of the study is aimed at identifying the reasons why accessibility concerns are overlooked. With the identified problems, we can propose solutions such as training conscientious professionals with adequate knowledge in human-computer interaction to work in the software development process.

Keywords: Software Engineering, Requirements Engineering, Software Developers, Human-Computer Interaction, Accessibility
A STUDY ON DISCOVERING ACCESSIBILITY ISSUES IN THE SOFTWARE DEVELOPMENT PROCESS

ABSTRACT: Accessibility in computing, when well applied, allows for the inclusion of users and the breaking of barriers in the interaction process. However, approaches to support accessibility validation are commonly unknown or ignored. The goal of this study is to understand accessibility issues considered in the software development process. This understanding is the basis for overcoming existing challenges. We want to find out why software developers disregard accessibility issues to research how to minimize this problem. We can find research papers addressing specific accessibility issues, such as mobile or web systems, but not broadly covering these issues. This research has carried out a systematic literature mapping, a questionnaire with people involved in the software development cycle, and a mapping of tools for automated accessibility tests. These three ways of collecting data help get information about accessibility issues in different but still related contexts. The results indicate that although there are approaches to support developers, the tools are still limited regarding the type of platform, type of disability considered, or accessibility guidelines applied. In addition, tools are often focused on evaluation after software development, which makes the adjustment process more costly. The study’s theoretical contribution is that accessibility rarely appears as a requirement in the early stages of software development. Despite recognizing its importance, according to the data collected, accessibility supposedly should be an intrinsic feature of the software, sometimes ignored by the development team for a set of reasons. Finally, the social contribution of the study is aimed at identifying the reasons why accessibility concerns are overlooked. With the identified problems, we can propose solutions such as training conscientious professionals with adequate knowledge in human-computer interaction to work in the software development process.

Keywords: Software Engineering, Requirements Engineering, Software Developers, Human-Computer Interaction, Accessibility

1 INTRODUCTION

Digital Information and Communication Technology has been used daily to assist in routine tasks and is firmly inserted in industrial, business, educational, and entertainment environments. Due to its great importance to society, it is necessary to innovate assistive technologies to generate digital inclusion and support people with disabilities in carrying out their activities. Then, when designing software, it is essential to think about accessible features and, bringing more usability to these users.

Accessibility is a criterion for the quality of use that provides people with the ability to interact without facing interface barriers [Barbosa et al., 2021]. In this context, accessibility applied to digital systems prevents users from having interaction difficulties and facilitates the performance of tasks and access to information.

Therefore, professionals responsible for software development must recognize accessibility as an essential criterion for social diversity and consider it in practice in their projects. On the one hand, making software accessible can be challenging since adding accessibility requirements to existing systems and user requirements influence the complexity, cost, and
time of their projects. On the other hand, methods and tools can favor these professionals’
decision-making, increasing user satisfaction and reducing complexity, cost, and time.

Given this scenario, the main goal of this work is to discover and understand accessibility
issues considered in the software development process. We divided the methodology of this
research into three main stages. The first stage of this work aimed to carry out a systematic
literature mapping (SLM) searching for studies focused on the state-of-the-art, considering
methods that teams adopt to approach accessibility requirements in software development.
In the second stage, we researched and mapped tools that aim to help verify accessibility in
software. Among the established research methods, we found the W3C website in which
we used the WCAG 2.0 and WCAG 2.1 guidelines filter for mapping tools and found 40
accessible tools. In the third stage of the work, we elaborated a questionnaire and applied
it to volunteers to analyze the relationship of software developers with the development of
accessible software, which obtained 132 responses.

The main contribution of this work is the research gaps on the inclusion of accessibility in
the software engineering process. The secondary contributions are in the results that indicate
that although there are approaches to support developers, the tools are still limited regarding
the type of platform, type of disability considered, or accessibility guidelines applied. There
is still no consensus on established tools widely used by software development companies.
The offer of tools based on artificial intelligence that helps software developers is still very
low, almost non-existent.

We were particularly interested in the Artificial Intelligence sub-area of computing, which
has been considered in including accessibility in the software development process. Since the
eighties, it has been possible to create disruptive technologies that approach the characteristics
of human reasoning, either through their structure or behavior (Russell & Norvig, 2002).
Working with artificial intelligence can be arduous due to the complexity of implemented
algorithms and the large amount of data they must process. However, its return on investment
is satisfactory in many applied cases, as it adds more automation in the execution of procedures.
Despite this, we did not find an expressive number of researches or tools in this area.

This paper is divided as follows: section 2 presents the related works; section 4 describes
an SLM to understand the state-of-the-art of accessibility validation approaches; section 5
presents a mapping of automated tools for accessibility validation; the section 6 presents
the results obtained through an online questionnaire applied to professionals of the software
development; in section 7 we deliver the results and discussions of the SLM, tool mapping
and online questionnaire; and, section 8 presents the final considerations and future works.

2 RELATED WORK

Tangarife and Mont’alvao (2006) carried out the first research in Brazil about accessibility
in software and the perception of web developers about the topic. The authors carried
out a survey answered by 68 volunteers of 70 who were invited by e-mail. Most of the
volunteers (94.11%) answered never had worked on accessibility or even in a project involving
accessibility.

Research by Freire, Goularte, and de Mattos Fortes (2007) aimed to identify techniques
for the construction of accessible software and what are the gaps faced by web development
professionals. For this, the researchers performed a systematic review, looking for tools,
methods, techniques, and processes used during the software development cycle that improves

https://www.w3.org/WAI/ER/tools/
the construction of accessible software, on the other hand, it also surveyed 605 web development professionals. The study identified that some of the difficulties faced by software development professionals were: the lack of technical support and lack of knowledge in accessibility recommendations.

Freire, Russo, and Fortes (2008) investigated accessibility awareness in web application development professionals. As a result, they applied an online questionnaire and collected 613 responses from web developers in all Brazilian states. As a result, the researchers realized that accessibility was not considered essential in developing web applications in academia, industry, or government, as only 19.9% of professionals performed activities for accessibility.

Dias, Fortes, Masiero, and Goularte (2010) presents an extension of the study by Freire et al. (2007). The objective was to identify techniques and methods used in the process of developing accessible software for the web. In this way, it carried out a state-of-the-art RSL from 2007 to 2010, where 65 research papers were analyzed. As an evaluation of the studies, the authors noticed that the number of researches on web accessibility has increased in several activities of the development process, such as requirements (9), project architecture (3), navigational projects (10), interface design (14), build (7), software test (13), user test (20), integration (2) and maintenance (4).

In contrast, the work by Alves (2011) applied an online questionnaire to analyze the difficulties presented by the Brazilian open-source software development community. In the end, the author created an assistive technology prototype developed through the suggestion of a free and accessible software development process adapted to the reality of these professionals.

The work by Antonelli, Rodrigues, Watanabe, and de Mattos Fortes (2018) presents the hypothesis that many websites are not accessible; therefore, it seeks to understand why web development professionals are not using accessibility requirements in their projects. Thus, it applied a questionnaire with web developers from all over Brazil, obtaining 404 valid responses. As a result, the authors realize that most professionals have never worked on building accessible web applications. However, 33.2% answered that they do not care about accessibility in their projects.

Santos, Salgado, and Fortes (2018) investigate the usability and accessibility aspects in the development process of games for seniors, thus indicating how and in what stages these aspects are used. Therefore, it uses the systematic literature mapping method to find answers to this problem in the state-of-the-art. The results show that the accessibility aspects are treated mainly in the implementation and evaluation stages of the project. On the other hand, it shows that the consideration of usability is greater than accessibility.

Rodrigues and Fortes (2019) presented a systematic review where they assessed the opportunities and barriers of accessibility in the development of tools for the Internet of Things. The research shows that accessibility has been considered in the process of developing Internet of Things tools, where 31 primary studies on assistive technologies were synthesized.

Paiva, Freire, and de Mattos Fortes (2021) was a continuation of research by Freire et al. Dias et al. (2007, 2010) where the researches investigated the state-of-the-art through systematic review, tools, methods, techniques, and processes used during the software development cycle that improves the construction of accessible software. The survey results show that from 2010 to 2021, there was an increase over previous years, particularly in software testing activities. On the other hand, the emphasis of the studies is on the processes of developing accessible software for partial or total visual impairment, and it is also important to develop primary studies that support software developers for other disabilities.

This research work differs from the previously presented by the union of an SLM, mapping.
of available tools in application repositories, and a questionnaire applied to professionals responsible for software development, which sought to understand the state-of-the-art and reality of the industry of development for software accessibility.

Regarding the SLM and mapping of available tools in application repositories, this study seeks to investigate approaches used to validate software requirements, applied to different contexts and people. Though, the SLM also looks for intelligent approaches, that is, tools and methods that use artificial intelligence concepts to validate accessibility requirements in software. This work and the other works presented are very important for the area of IHC, Software Engineering and also for people with disabilities or reduced mobility, as they present the evolutions in the development of accessible software in different contexts and the barriers that still need to be overcome for the social and digital inclusion of people, access to information and communication through education, health, mobility, housing, among others.

3 RESEARCH METHODOLOGY

We conducted exploratory research based on data collected from scientific literature, websites, smartphone app stores, and questionnaires with people involved in the software development process. We also divided the methodology of this research into three main stages. Each of the data collection steps has a specific objective.

The first stage of this work aimed to carry out a systematic literature mapping (SLM) searching for studies focused on state-of-the-art, considering methods that teams adopt to approach accessibility requirements in software development. We selected and presented 16 studies as a result of this research. The specific goal of the mapping was to analyze if there are methods, tools, processes, and other approaches for evaluating accessibility requirements and how they are helping software developers.

In the second stage, we researched and mapped tools that aim to help verify accessibility in software. Among the established research methods, we found the W3C website in which we used the WCAG 2.0 and WCAG 2.1 guidelines filter for mapping tools and found 40 accessible tools among websites, software for windows and Linux, and browser plugins. The specific goal of this search was to know the characteristics of existing tools and understand if they meet the needs of software developers.

In the third stage of the work, we elaborated a questionnaire and applied it to volunteers to analyze the relationship of software developers with the development of accessible software, which obtained 132 responses. The specific goal of the questionnaire was to understand the scenario of software development in Brazil concerning issues about accessibility.

It is noteworthy that the primary goal of this work is to discover and understand accessibility issues considered in the software development process. We hope that the union of these collection methods will provide a broad knowledge of different points of view: scientific research related to the topic, the tools available for use, and how people involved in the software development process deal with usability issues.

4 SYSTEMATIC LITERATURE MAPPING

The Systematic Literature Mapping (SLM) is a scientific research methodology conceived in the medical field and later adapted for computing, initially for software engineering. According to Kitchenham et al. (2009) there are two strategic and organized means of
observation, systematic literature review (SLR) and SLM. The SLM is intended to seek primary studies to collect and analyze scientific evidence to generate general knowledge about a particular topic, while the RSL is more specific in exhaustively evaluating the identified evidence.

The SLM method was used at this research stage to find approaches supporting software development teams in the accessibility requirements validation process. In this context, it was also observed whether the approaches had traditional or intelligent characteristics. Figure 1 presents eight activities we used to build this mapping and which were grouped into three essential steps: planning, conducting, and reporting (Kitchenham et al., 2009).

![Figure 1: Steps and activities of a systematic mapping. Source: Kitchenham et al. (2009)](image)

4.1 Systematic mapping planning

The first step of the planning was to identify the need to produce a systematic review or mapping. We searched for secondary studies that involved accessibility issues applied to the software development process. As a result, the secondary studies Freire et al.; Dias et al.; Santos et al.; Rodrigues and Fortes; Paiva et al. (2007; 2010; 2018; 2019; 2021) were found, considered as related works. When evaluating the studies mentioned, we identified the need to create a systematic mapping to collect and categorize the works related to intelligent approaches that support software development teams in validating accessibility requirements.

4.1.1 Main search question

In this activity, the main research question for the MSL was built. Thus, a general question was defined to analyze if there are methods, tools, processes, and other approaches for assessing accessibility requirements and how they are helping software developers. In this context, the main research question is presented below:

- RQ1 - How can approaches (methods and tools) assist developers in validating software accessibility requirements?

4.1.2 Secondary research questions

Four important secondary research questions for the MSL were considered. In general, each secondary research question aims to collect information about other specific problems
relevant to the understanding of the returned studies and support the construction of possible new primary or secondary studies.

Through the first secondary research question (RQ2), we sought to understand how the approaches mentioned in the works have improved the accessibility of people with disabilities or reduced mobility. The second secondary research question (RQ3) aims to identify if and which guidelines are used in the approaches.

The third secondary research question (RQ4) aimed to investigate whether these approaches are intelligent, that is, whether they have used artificial intelligence to support the accessibility assessment during the software development process. The fourth and last secondary research question (RQ5) aimed to collect artificial intelligence concepts used in the primary studies and how their authors used them. The two research questions are important to know whether accessibility research has considered artificial intelligence to support decision-making on aspects of accessibility or whether traditional assessment methods prevail.

- **RQ2** - How does the use of approaches to validate accessibility requirements in software improve access for people with disabilities?
- **RQ3** - What software accessibility guidelines are used by approaches that validate accessibility requirements?
- **RQ4** - Are the approaches used considered intelligent?
- **RQ5** - What artificial intelligence concepts are used in building intelligent approaches to validate accessibility requirements?

### 4.2 Systematic mapping conducting

This step of mapping was responsible for creating the search string, choosing the bases, defining the inclusion and exclusion criteria, selecting and evaluating the primary studies. After performing all the steps mentioned, it was possible to have the list of primary studies selected to answer the questions defined in the MSL planning stage and presented previously.

#### 4.2.1 Definition of research repositories

The scientific databases “ACM Digital Library” and “IEEE Xplorer” were selected due to their efficient search engines [Buchinger, Cavalcanti, & Hounsell, 2014]. Other criteria used were the availability of access through the platform of the federated academic community (CAFE), the Coordination for the Improvement of Higher Education Personnel (CAPES) in partnership with the National Education and Research Network (RNP), and being scientific reference bases for computer science conferences, journals, and research groups. The two databases enabled automated searches through the search field compatible with search strings; however, each search string had to be adapted to the syntax compatible with the search engines of their platforms.

#### 4.2.2 Search string

The search string was constructed using the PICO method, where the Population (P), Intervention (I), Comparison (C), and Outcome of interest (O) of the search were described [Kitchenham et al., 2009]. Table [I] presents the application of the PICO method and the results obtained in the research.
Two tests were performed and, consequently, two modifications in keywords or synonyms. With that, we obtained the suitable search string, being: “(“software development” OR “mobile development” OR “web development” OR “application development”) AND (“accessibility”) AND (“testing” OR “evaluation” OR “verification” OR “validation”)”.

### 4.2.3 Inclusion and exclusion criteria

We developed one inclusion criteria (IC) and four exclusion criteria (EC) during the construction of the research protocol. Both types of criteria were used to bring more accuracy to the study list returned by search engines. The criteria produced were:

- **IC1** - Returned works must contain the search string keywords in their abstracts;
- **EC1** - Secondary studies should not be considered. In other words, it is mandatory to remove from the list of returned works studies that are SLM or SLR;
- **EC2** - Works must be written in English or Brazilian Portuguese;
- **EC3** - Only works in which thematic area is software engineering or human-computer interaction will be considered;
- **EC4** - Works found in more than one repository should be considered only once.

### 4.2.4 Selection of primary studies

In the first selection cycle of primary studies, the search string was used to return works containing the keywords in the text, collecting 2627 studies from the academic bases, 2595 from ACM, and 32 from the IEEE. Table 2 presents the results obtained through the first research selection cycle.

<table>
<thead>
<tr>
<th>Base</th>
<th>Studies returned</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACM</td>
<td>2595</td>
</tr>
<tr>
<td>IEEE</td>
<td>32</td>
</tr>
<tr>
<td>Total</td>
<td>2627</td>
</tr>
</tbody>
</table>
4.2.5 Evaluation of primary studies

In the second cycle, we applied the inclusion criterion, where we searched for works that contained the keywords in their abstracts, this time collecting 13 studies from ACM and 12 studies from IEEE. Finally, in the third cycle, applying the exclusion criteria, eight works were found in each database, totaling 16 studies selected from the SLM (see Table 3). The first inclusion criterion (IC1) and the second exclusion criterion were automatically applied through the filters available in the search engines of the scientific databases. The other exclusion criteria (EC1, EC3, and EC4) were manually applied.

<table>
<thead>
<tr>
<th>Base</th>
<th>Inclusion criteria</th>
<th>Exclusion criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACM</td>
<td>13</td>
<td>8</td>
</tr>
<tr>
<td>IEEE</td>
<td>12</td>
<td>8</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>25</strong></td>
<td><strong>16</strong></td>
</tr>
</tbody>
</table>

4.3 Systematic mapping results

After filtering through the inclusion and exclusion criteria, we started the data extraction and synthesis activities and summarizing the results obtained. Table 4 presents the 16 articles selected for extraction and analysis of important data for this research.

<table>
<thead>
<tr>
<th>Id</th>
<th>Title</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>Application of Traditional Software Testing Methodologies to Web Accessibility</td>
<td>Shelly and Barta [2010]</td>
</tr>
<tr>
<td>A2</td>
<td>Evaluating a Usability, User Experience and Accessibility Checklist for Assistive Technologies for Deaf People in a Context of Mobile Applications</td>
<td>de Godoi, García, and Valentim [2020]</td>
</tr>
<tr>
<td>A3</td>
<td>Fona: Quantitative Metric to Measure Focus Navigation on Rich Internet Applications</td>
<td>Watanabe, Dias, and Fortes [2015]</td>
</tr>
<tr>
<td>A5</td>
<td>Localizing Accessibility of Text Alternatives for Visual Content in Multilingual Websites</td>
<td>Vázquez [2013]</td>
</tr>
<tr>
<td>A6</td>
<td>Unlocking the Potential of Web Localizers as Contributors to Image Accessibility: What Do Evaluation Tools Have to Offer?</td>
<td>Vázquez [2015]</td>
</tr>
<tr>
<td>A7</td>
<td>Using Visual Intelligence to Automate Maintenance Task Guidance and Monitoring on a Head-Mounted Display</td>
<td>Ng et al.</td>
</tr>
<tr>
<td>A8</td>
<td>Web Accessibility Testing for Singapore Government E-Services</td>
<td>Lim, Chua, Yang, Tan, and Chai [2020]</td>
</tr>
<tr>
<td>A9</td>
<td>Accessibility in an educational software system: Experiences and Design Tips</td>
<td>Pathauer and Rao [2019]</td>
</tr>
<tr>
<td>A10</td>
<td>Adapting SCRUM Methodology to Develop Accessible Web Sites</td>
<td>Romero, Muir, Rodríguez, Gómez, and Chacón [2019]</td>
</tr>
<tr>
<td>A11</td>
<td>Crowd Sourced Evaluation Process for Mobile Learning Application Quality</td>
<td>Khan, Al-Khanjar, and Sarrab [2017]</td>
</tr>
<tr>
<td>A12</td>
<td>Developer Recommendation for Crowdsourced Software Development Tasks</td>
<td>Mao, Yang, Wang, Jia, and Harman [2015]</td>
</tr>
<tr>
<td>A13</td>
<td>Evaluating the Effectiveness of BEN in Localizing Different Types of Software Fault</td>
<td>Chandrasekaran, Ghandehari, Lei, Käcker, and Kuhn [2016]</td>
</tr>
<tr>
<td>A14</td>
<td>Improving project planning/tracking for student software engineering projects through SOPPTs</td>
<td>Zhang, Zage, and Zage [2003]</td>
</tr>
<tr>
<td>A15</td>
<td>Technology transfer issues for formal methods of software specification</td>
<td>Abernethy, Kelly, Sobel, Kiper, and Powell [2000]</td>
</tr>
</tbody>
</table>
4.3.1 Data extraction and synthesis

In this last step, we categorized the extracted data. As for the year of publication, it is worth noting that three studies were published in 2015 and three studies in 2019, being the years with the highest number of publications. In 2012 and 2020 there were two publications (see Figure 2).

![Figure 2: Year and repository of selected works.](image)

Source: Authors

As for the countries of the first authors, the USA has the highest concentration of articles, with five publications, followed by Singapore, Brazil, United Kingdom, and Switzerland with two published works each (see Figure 3).

![Figure 3: Countries of selected works.](image)

Source: Authors

Considering the places where the selected works were published, it is worth mentioning the event “W4A”, which had four works among the selected ones. In addition to this, the event “CSEE&T” had 2 works among those selected (see Table 5).
4.3.2 Summary of results

a) RQ1: How can approaches (methods and tools) assist developers in validating software accessibility requirements?

For RQ1, the mapping showed as an answer how the approaches help developers validate software accessibility requirements. It is important to emphasize that the works whose target audience were people with disabilities and presented approaches to software requirements validation were considered satisfactory answers.

Three types of approaches were identified: processes, methods, and tools. The methods identified in the studies are techniques used by developers to cognitively and manually analyze accessibility issues. Tools are approaches that support developers in an automated way. On the other hand, processes are approaches presented by some studies and represent a set of methods and tools for the development of accessible software.

Table 5 presents a summary of each article with the extraction of their approaches. Therefore, this list of articles presents one methodology, four processes, and four tools that support developers in validating accessibility requirements.

| Local Conferences and Journals where selected works were published (Authors, 2021). |
|-----------------|-----------------|
| International Cross Disciplinary Conference on Web Accessibility (W4A) | ACM | A1, A4, A6, A8 |
| Conference on Software Engineering Education & Training (CSEE&T) | IEEE | A14, A15 |
| ACM Transactions on the Web (TWEB) | ACM | A3 |
| Brazilian Symposium on Human Factors in Computing Systems (IHC) | ACM | A2 |
| Frontiers in Education (FIE) | IEEE | A9 |
| Frontiers of Information Technology (FIT) | IEEE | A16 |
| International Conference on Inclusive Technologies and Education (CONTIE) | IEEE | A10 |
| International Conference on Information Systems Engineering (ICISE) | IEEE | A11 |
| International Conference on Robotics and Artificial Intelligence (ICRAI) | ACM | A7 |
| International Workshop on Service-Oriented System Engineering (SOSE) | IEEE | A12 |
| International Conference on Software Testing Verification and Validation Workshop (ICSTW) | IEEE | A13 |
| Special Interest Group on Accessibility and Computing (SIGACCESS) | ACM | A5 |

Table 6 presents a summary of each article with the extraction of their approaches. Therefore, this list of articles presents one methodology, four processes, and four tools that support developers in validating accessibility requirements.

<table>
<thead>
<tr>
<th>Studies</th>
<th>Approaches</th>
<th>Studies</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>Process</td>
<td>User Testing, Traditional Software Testing and Auditing</td>
</tr>
<tr>
<td>A2</td>
<td>Method</td>
<td>Checklist to assess accessibility, usability and experience of deaf users in mobile applications (UXAC-DAT)</td>
</tr>
<tr>
<td>A3</td>
<td>Tool</td>
<td>Quantitative metrics for automated assessment (Speech)</td>
</tr>
<tr>
<td>A4</td>
<td>Tool</td>
<td>Automated website validator (AChecker)</td>
</tr>
<tr>
<td>A6</td>
<td>Tool</td>
<td>Automated text validator (Acrolinx) and website (aDesigner)</td>
</tr>
<tr>
<td>A8</td>
<td>Process</td>
<td>Agile methodology, test tool (Ax from Deque Systems and Selenium), and continuous integration platform (Secure Hybrid Integration Platform)</td>
</tr>
<tr>
<td>A9</td>
<td>Process</td>
<td>Agile methodology, automated website validator (WAVE) and accessibility specialist</td>
</tr>
<tr>
<td>A10</td>
<td>Process</td>
<td>Scrum Methodology: Planning, Backlog, Sprint (design, code, accessibility test, accessibility review, test, deploy)</td>
</tr>
<tr>
<td>A16</td>
<td>Tool</td>
<td>Automated Website Validator (Total Validator and Functional Accessibility Evaluator 1.0.2)</td>
</tr>
</tbody>
</table>

Processes

Shelly and Barta (2010) reinforce the use of a set of traditional development methods for assessing web accessibility. They indicate that usability testing, automated assessment, and auditing approaches assist developers in planning and assuring a certain quality level of
accessibility requirements. In addition, using the approach brings experience to the team in designing new accessible products and services.

On the other hand, Fathauer and Rao; Romero et al. (2019; 2019) indicate approaches aimed at the agile software development process. According to Fathauer and Rao (2019), agile methodologies need the support of an accessibility professional to generate a process where each sprint is continuously evaluated. In this context, these professionals are responsible for the semantics and user experience in accessible software, while automated tools support the evaluation of accessible user interfaces.

The approach presented by Romero et al. (2019) provides accessibility management at each sprint, so IT professionals are responsible for planning, coding, and evaluating accessibility requirements at each iteration, allowing for the consideration of the time needed for the development of the proposal.

The approach of Lim et al. (2020) is based on a set of methods and tools adapted to accessibility. The automated assessment tool uses the open-source Ax code library, which allows for flexibility and customization of accessibility inspection according to the local scenario, as in the case of the study applied in Singapore. As these tools move toward automated accessibility testing through continuous integration platforms, they give developers collective responsibility for assessing accessibility features and free them from manual and advanced analytics that pipelines can automate.

Methods

The research by de Godoi et al. (2020) presents the UUXAC-DAT method, which is a checklist used by developers to indicate problems in accessibility requirements for mobile applications. Therefore, it is possible to assess whether an application is inclusive or not. The approach also provides a complete assessment of the interface, which can easily find different development errors. As much as the method is for evaluating mobile applications, it has easy adaptability for evaluating accessibility, usability, and user experiences in other contexts, such as games, web applications, and desktop applications.

Tools

The aChecker tool, used by Gilbertson and Machin (2012), provides support to developers in validating requirements with general guidelines for internet accessibility. Moreover, since this is an automated tool, it saves time and cognitive load in checking accessibility. Bakhsh and Mehmood (2012) indicates that automated approaches do not generate new project costs, facilitate implementation, ease the cognitive load needed to identify accessibility issues, and help development teams follow universal accessibility guidelines in the web.

The approaches used by Vázquez (2015) help web developers and translators in the validation of image captions of web pages with more than one language, which reduces the evaluation time and also the complexity of implementing more than one language. Watanabe et al. (2015) introduces Fona, which is a tool used to inspect Focus Navigation requirements. The approach indicates to web development teams the number of interface flaws that can be resolved through keyboard interaction.

b) RQ2: How does the use of approaches to validate accessibility requirements in software improve access for people with disabilities?

For RQ2, we analyzed whether the use of approaches to the validation of accessibility requirements has improved accessibility. Thus, the results of this research question were
divided into four categories of service improvement, namely: information and communication, government, education, and business. The access to information and communication category was considered to studies that did not define a specific context for evaluation through processes, methods, or techniques.

Table 7 presents a summary of the results obtained for the second research question; therefore, four studies propose improvements in the access to information and communication services (one for the deaf and three for people with disabilities in general), three to government services, one to educational services and, finally, one to services of private organizations. In another context, it is clear that the rise is in the evaluation of web platforms, with a total of eight studies and only one for improvements in mobile applications.

Table 7: Summary of answers found for RQ2 (Authors, 2021).

<table>
<thead>
<tr>
<th>Study</th>
<th>Context and Public</th>
<th>Platforms</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1, A3 and A6</td>
<td>Improve access to information and communication for people with disabilities</td>
<td>Web</td>
</tr>
<tr>
<td>A8, A10 and A16</td>
<td>Improve access to information and communication for people with disabilities in public organization services</td>
<td>Web</td>
</tr>
<tr>
<td>A2</td>
<td>Improve access to information and communication for deaf people</td>
<td>Mobile</td>
</tr>
<tr>
<td>A9</td>
<td>Improve access to information and communication for students with and without disabilities in educational services</td>
<td>Web</td>
</tr>
<tr>
<td>A4</td>
<td>Improved access to information and communication of PCDs in services of private organizations</td>
<td>Web</td>
</tr>
</tbody>
</table>

Access to information and communication

With the studies of Shelly and Barta (2010) we noticed that the combination of usability testing, automated validation, and heuristic evaluations bring more quality to the access to information and communication of people with disabilities through web applications. The combination of Acrolinx and aDesigne presented by Vázquez (2015) provided higher quality in alternative texts of web page media content, such as images, as they improve the translation of these content for people with visual impairments.

After analyzing the work of Watanabe et al. (2015) we noticed that for people with disabilities, the use of interfaces with pointers is not the most suitable. Thus, Fona assesses whether the functions of web applications provide interactivity through interfaces other than the mouse, making pages more accessible for people with reduced mobility. The work of de Godoi et al. (2020) focused on evaluating a mobile application for the deaf. Therefore, the assessment of assistive technologies through a checklist meets the needs of the deaf community, which improves their satisfaction and use, in addition to considering their experiences of using mobile applications.

Access to public organizations services

The work of Bakhsh and Mehmood (2012) has indicated that automated accessibility assessment can provide improvements in the interaction of people with disabilities on Pakistani government platforms, providing them with inclusive access to essential and available services by the state through the internet, such as requesting and updating assets and personal documents.

On the one hand, the use of Scrum adapted to the planning, coding, and evaluation of accessibility has generated satisfaction in the use of products and services from the government.
of Costa Rica for people with disabilities since the features developed are evaluated at each sprint and not only when a user alerts a problem, thus making the requirements more accurate with the users’ needs (Romero et al., 2019). On the other hand, the adaptation of the accessible software development and evaluation process has also improved the access of people with visual and physical disabilities to Singapore government services, as their needs were elicited, specific, and considered during the evaluation of a platform web (Lim et al., 2020).

**Access to educational services**

Combining the agile method and assessment of accessibility requirements generate fast educational products and services that enhance the teaching and learning of students with or without disabilities. In this context, the plugin developed by Fathauer and Rao (2019) provided users with the facility to forward their activities on the platform, sending them quickly close to the maximum submission closing time.

**Access to services from private organizations**

The study by Gilbertson and Machin (2012) made it possible to evaluate various interface aspects for access improvements for people with disabilities in companies that provide software development services. However, what stood out the most was the problems of alternative subtitles for non-textual resources, identified by aChecker. Thus, we noticed that the approach is a satisfactory option to indicate the problems of people with visual impairment or low vision and, thus, improve these people’s access to the pages of software service providers.

c) **RQ3: What software accessibility guidelines are used by approaches that validate accessibility requirements?**

For RQ3, we analyzed which accessibility guidelines the studies followed. We noticed that WCAG was the most cited guideline among the works, with a total of 6 uses. Also, the Axe 3.4 and WAI-ARIA guidelines were cited only once. However, eight papers do not mention which guidelines they used.

d) **RQ4: Are the approaches used considered intelligent?**

For RQ4 and RQ5, we analyzed only two works that presented intelligent approaches, that is, processes, methods and tools that use the concepts of artificial intelligence for decision making. The works defining smart approaches were Mao et al.; Ng et al. (2015; 2019); however, the accessibility presented does not target people with disabilities.

e) **RQ5: What artificial intelligence concepts are used in building intelligent approaches to validate accessibility requirements?**

The work by Ng et al. (2019) uses computer vision and machine learning in its framework to support the user during maintenance tasks. In addition, the work by Mao et al. (2015) uses supervised learning and decision tree in the tool called “CrowdRex”, which suggests software developers for the development of a task in an automated way, based on the developer’s history. However, as stated in the fourth research question, the accessibility presented by these two studies does not target people with disabilities.
5  MAPPING OF AVAILABLE ACCESSIBILITY TOOLS

At this stage of the research, we searched for automated inspection tools that can be used during the development cycle, supporting developers in validating accessibility requirements. We used Google Play Store, Google Search Engine, and Chrome Web Store and Firefox Browser add-ons as bases for identifying these tools.

As a result of the searches on the platforms mentioned, we found two functional tools in Google Play Store using the keywords: “accessibility for android”, “accessibility assessment”, and “accessibility”, in which it was possible to evaluate applications and smartphone functionalities, the tools showed the results of the analysis through error instructions, alerts, and tweaking tips to make the application accessible according to the selected guideline.

Table 8: Tool mapping (Authors, 2021).

<table>
<thead>
<tr>
<th>Name</th>
<th>Category</th>
<th>Tool description</th>
<th>Type of Service</th>
<th>Released In</th>
</tr>
</thead>
<tbody>
<tr>
<td>AX by Respect</td>
<td>Web</td>
<td>Integrates testing tool to verify accessibility</td>
<td>Free</td>
<td>12/17/2017</td>
</tr>
<tr>
<td>AX by Respect</td>
<td>Web</td>
<td>Color contrast accessibility validator</td>
<td>Free</td>
<td>02/27/2021</td>
</tr>
<tr>
<td>Access Continuum</td>
<td>Web</td>
<td>Accessibility test integration SDK</td>
<td>Free</td>
<td>05/25/2018</td>
</tr>
<tr>
<td>Access Analytics</td>
<td>Web</td>
<td>Performs accessibility adjustment as needed</td>
<td>Free</td>
<td>05/29/2018</td>
</tr>
<tr>
<td>Accessibility Insights</td>
<td>Plugins</td>
<td>Helps to find and fix accessibility problems on websites</td>
<td>Free</td>
<td>03/19/2019</td>
</tr>
<tr>
<td>Accessibility checker</td>
<td>Linux</td>
<td>Integrated testing tool to verify accessibility</td>
<td>Free</td>
<td>05/18/2020</td>
</tr>
<tr>
<td>Accessible Brand Colors</td>
<td>Web</td>
<td>Accessible Brand Colors</td>
<td>Free</td>
<td>02/28/2019</td>
</tr>
<tr>
<td>Acessi.org</td>
<td>Web</td>
<td>Verifies website accessibility according to the rules of the WCAG</td>
<td>Free</td>
<td>01/01/2021</td>
</tr>
<tr>
<td>Accessibility Engine (axe)</td>
<td>Android</td>
<td>Android application &amp; Accessibility analysis tool for Android</td>
<td>Free</td>
<td>Not informed</td>
</tr>
<tr>
<td>Aditus</td>
<td>Web</td>
<td>Test buttons and links according to the WCAG</td>
<td>Free</td>
<td>09/10/2019</td>
</tr>
<tr>
<td>ANDI</td>
<td>Plugins</td>
<td>Accessibility testing tool</td>
<td>Free</td>
<td>08/23/2017</td>
</tr>
<tr>
<td>ARC Toolkit</td>
<td>Plugins</td>
<td>Accessibility testing tool</td>
<td>Free</td>
<td>05/16/2019</td>
</tr>
<tr>
<td>Axes-web</td>
<td>Web</td>
<td>Website accessibility evaluation</td>
<td>Free</td>
<td>Not informed</td>
</tr>
<tr>
<td>AssistivLabs</td>
<td>Web</td>
<td>Test the experience with assistive technologies</td>
<td>Free and Paid</td>
<td>11/10/2020</td>
</tr>
<tr>
<td>Ax Auditor</td>
<td>Web</td>
<td>Accessibility testing tool</td>
<td>Free</td>
<td>12/02/2016</td>
</tr>
<tr>
<td>Axe core</td>
<td>Linux</td>
<td>HTML-based accessibility testing</td>
<td>Free</td>
<td>01/10/2015</td>
</tr>
<tr>
<td>Axe dev Tolls</td>
<td>Plugins</td>
<td>Accessibility Check and Fix Tool</td>
<td>Free</td>
<td>10/02/2017</td>
</tr>
<tr>
<td>Axe dev Tolls</td>
<td>Plugins</td>
<td>Website accessibility check</td>
<td>Free</td>
<td>06/10/2015</td>
</tr>
<tr>
<td>Colour Contrast Checker</td>
<td>Web</td>
<td>Color contrast test</td>
<td>Free</td>
<td>10/04/2020</td>
</tr>
<tr>
<td>Dyno Mapper</td>
<td>Web</td>
<td>Website accessibility check</td>
<td>Free</td>
<td>03/15/2019</td>
</tr>
<tr>
<td>EqualWeb</td>
<td>Web</td>
<td>Website accessibility check</td>
<td>Free</td>
<td>02/01/2014</td>
</tr>
<tr>
<td>Expertie</td>
<td>Web</td>
<td>Accessibility testing tool</td>
<td>Free</td>
<td>07/27/2020</td>
</tr>
<tr>
<td>HtmlValidator</td>
<td>Web</td>
<td>HTML validator</td>
<td>Free</td>
<td>10/16/2019</td>
</tr>
<tr>
<td>Hurg</td>
<td>Web</td>
<td>Accessibility testing according to the WCAG</td>
<td>Free</td>
<td>09/23/2020</td>
</tr>
<tr>
<td>Image-color</td>
<td>Web</td>
<td>Color contrast checker</td>
<td>Free</td>
<td>05/26/2019</td>
</tr>
<tr>
<td>Infosys accessibility</td>
<td>Document</td>
<td>Document accessibility testing tool</td>
<td>Free</td>
<td>06/01/2020</td>
</tr>
<tr>
<td>Karma</td>
<td>Linux</td>
<td>JavaScript accessibility testing tool</td>
<td>Free</td>
<td>05/18/2020</td>
</tr>
<tr>
<td>Lighthouse</td>
<td>Plugins</td>
<td>Website accessibility check</td>
<td>Free</td>
<td>Not informed</td>
</tr>
<tr>
<td>MAUVE</td>
<td>Web</td>
<td>Website accessibility check</td>
<td>Free</td>
<td>01/18/2019</td>
</tr>
<tr>
<td>Monsudo</td>
<td>Web</td>
<td>Color contrast checker</td>
<td>Free</td>
<td>07/11/2019</td>
</tr>
<tr>
<td>MSU NatSci</td>
<td>Linux</td>
<td>Check for possible accessibility violation</td>
<td>Free</td>
<td>08/10/2019</td>
</tr>
<tr>
<td>QualWeb</td>
<td>Web</td>
<td>WCAG-based automated accessibility checker</td>
<td>Free</td>
<td>01/01/2018</td>
</tr>
<tr>
<td>Scanner de acessibilidade</td>
<td>Android</td>
<td>Quick accessibility check yool</td>
<td>Free</td>
<td>Not informed</td>
</tr>
<tr>
<td>Tawdx</td>
<td>Web</td>
<td>Website accessibility check</td>
<td>Free</td>
<td>02/01/2000</td>
</tr>
<tr>
<td>UsableNet</td>
<td>Web</td>
<td>Accessibility Testing tool</td>
<td>Free</td>
<td>08/01/2016</td>
</tr>
<tr>
<td>UserWay</td>
<td>Web</td>
<td>Real-time accessibility checking and monitoring</td>
<td>Free and Paid</td>
<td>05/01/2020</td>
</tr>
<tr>
<td>Usan</td>
<td>Web</td>
<td>WCAG 2.1-based design process tool</td>
<td>Free</td>
<td>12/01/2018</td>
</tr>
<tr>
<td>Wave accessibility</td>
<td>Web</td>
<td>Website accessibility check</td>
<td>Free</td>
<td>01/01/2014</td>
</tr>
<tr>
<td>WebAccessibility</td>
<td>Web</td>
<td>LevelAccess’s Accessibility Check Tool</td>
<td>Free</td>
<td>12/19/2017</td>
</tr>
</tbody>
</table>

On Google Search, we found a website, “Web Accessibility Evaluation Tools List” (W3C [2021] ), with a list of web accessibility tools mapped and verified from April to May 2021. The website also provided a filtering capability and allowed the tools to be searched using several filters. Among the available filters, we used the guideline filter, selecting WCAG 2.0 and WCAG 2.1, as they are more recent, and found 55 tools. From the tools we found, 25 are active and met the research objective, nine are active and only focused on color blindness, seven are inaccessible (page not found) and met the research objective according to the tool description, nine are inaccessible and do not meet the research objective, and the remaining
five only updates to previously found tools. For the search on the Chrome Web Store and Firefox Browser add-ons, the tools found were the same tools already found on the website mentioned above. The tools found in the mapping carried out were compiled and are shown in Table 8, organized by name, category, description, and whether they are paid or free.

Among the tools found and accessible, we observed that 26 are accessed via web browser, six are browser plugins, two android applications, and desktop tools, five are Linux tools, and one document with information on how the company develops its accessibility tests. The other links were updated versions to already published tools, or we could not access them because they were disabled. Then, we had the effective mapping of 40 tools, of which 36 have free access, two have paid versions, and two have paid and free versions (see Figure 4).

![Figure 4: Tool mapping overview. Source: Authors](image)

We also note that the most frequent year of release of tools is 2019 and 2020, with ten tools released in 2019 and nine in 2022. So far, in 2021, only two tools had been released. The others tools found are in the range of 17 years. In 2000, only one tool was released, with an interval of 14 years until 2014 when two tools were released, followed by 2015 and 2016 with two tools per year. In 2017 and 2018, four tools were released per year, and four did not provide the release year (see Figure 5).

![Figure 5: Years of release tools. Source: Authors](image)

Among the tools found in the mapping, we performed tests to understand the accessibility check analysis guidance. We check what the tool can provide information and whether it
is objective, whether it can show possible solutions to make the software accessible, and whether it complies with the WGAC rules. Therefore, we note that among the functional tools found, few encompass the three strands of targeting. Typically, it fits into sites where a text box is shown as the initial screen for entering an URL to assess accessibility. Also, in their results, they have presented the problems encountered and suggestions for solutions.

Regarding the installable Linux tools, in general, the steps for installation and verification are longer, which may imply little developer interest in using the tool if the goal is to use a tool with the fewest possible steps to check the accessibility of the software. Following the same line of reasoning, web companies focused on accessibility verification prioritize requesting a registration to send reports or request a reservation for mapping the URL informed, leading to a longer time for validation of this verification.

Depending on the developer’s interest in finding tools to verify their software, the possibilities mentioned may imply a low effective demand for this type of verification. Even though since 2000, it has been mandatory for software and websites to be accessible, according to the law of Brazil (2002).

6 SURVEY WITH SOFTWARE DEVELOPMENT PROFESSIONALS

We created a questionnaire and made it available to volunteers in order to understand the current scenario of the inclusion of accessibility concerns in the software development process. The questionnaire had 25 questions aimed at developers and managers of development teams. According to the position and previous experiences with accessible development, we presented different questions specific to their profile.

The questionnaire was answered by 132 volunteers, mostly men (83.3%) and aged between 20 and 39 years (90.2%). As for the type of organization in which volunteers work, the highest percentage (86.4%) works in the Corporate, Government, or Academic settings, with 56.1% concentrated in the Corporate setting. The number of employees of the organizations is evenly distributed between 0 to 20, 21 to 100, 101 to 500, and over 500 employees (about 25% in each group). Additionally, most participants (61.4%) exercise the position of Software Developer, followed by Manager with 14.4% and the remaining answers split between other positions.

Of the 132 volunteers, 68 (51.5%) answered that they had already worked with accessibility at the beginning of the questionnaire. For these people, we asked specific questions about the development of accessible software. The first question sought to understand why they had worked with accessibility. For this question, 51.5% answered that it was a requirement of the project, 40% indicated that there was personal interest, 29.5% answered having implemented accessibility due to public recommendations, and the rest were unable to inform the reason.

In this context, 70% of the volunteers answered they do not have direct contact with any person with any disability; of the 30% who answered to have contact, 3.8% have, themselves, a disability. Only 50% have worked on a project that considered accessibility aspects, although more than 95% consider it important to adopt concerns with accessibility in software projects (see Figure 6).

We also asked the volunteers if the projects they participated in had any accessibility concerns, even if not implemented in the final product. As Figure 7 shows, 37.9% answered there was no concern, 19% answered that there was no concern, but there are plans to include it in the future, and the rest (43.8%) answered that there were accessibility concerns in some or all projects they had participated.
Participants also answered about the target demographic for which they have already developed projects concerned with accessibility, and Figure 8 presents the results obtained. The most significant number of responses concentrated between visual impairment, older people, and hearing impairment, with (45.5%, 25%, and 22.7%, respectively), among others shown in the same figure. Another 37.9% indicated that they had never worked with accessibility.

In another question, volunteers answered about their familiarity with the Brazilian laws that enforce the implementation of accessibility in software, 50% indicated not even knowing about the existence of these laws. At the same time, 33.3% have heard about the laws but have no in-depth knowledge, and only 16.7% know the laws and their implications. Another question was about what the volunteers considered the biggest challenges to insert accessibility concerns into software projects; Figure 9 presents a chart with the participants’ answers. The four categories that concentrated most of the answers were: not being a requirement for the customer (68.2%), lack of training (53%), lack of a specialist in the team (51.5%), and not being a requirement for the organization (47%).

Concerning the organization they work for, the volunteers also answered about the number of accessibility specialists. Most respondents (66.7%) indicated that there was no specialist in their organization, 12.1% were not sure of the answer, the remaining 21%, approximately, indicated that there were between one, two, three or more software accessibility specialists, as shown in Figure 10. The questionnaire also asked about which platforms the volunteers had already developed for with accessibility concerns: 70% answered web applications, 33% answered mobile applications, and 20% answered desktop applications. The volunteers also answered about which accessibility guidelines they were familiar with, Figure 11 summarizes...
the responses, with the most prominent guidelines being: WCAG (42.2%) and eMAG (31.1%). Furthermore, 31% of the volunteers responded that they did not know any accessibility guidelines.

Regarding the tools for accessible development used by the volunteers, the most frequent (aggregating 50% of the answers) were: Accessibility Insights (16.2%), Lighthouse (11.8%), Wave Accessibility (7.4%), Ases-web (5.9%), E-scanner (2.9%), Tawdis (2.9%), and VLibras (2.9%). The other half of the responses indicated that they used less popular tools or had never used any tools. We also asked at what stage of development the volunteers used these tools: 38% indicated using them in the testing phase, 38% indicated using them in the coding phase, 23.5% indicated using them in the deployment phase, 22.1% in the requirements gathering and analysis phase, 14.7% in the maintenance phase and 13.2% in the diagnostics phase.

Finally, we had the volunteers who answered at the beginning of the questionnaire to have never worked with accessibility explain why it had never happened. Of the volunteers, 86% answered that it was not a requirement in any project, 53.1% responded that the projects they work on do not deal with accessibility, 9.4% did not know how to answer, 2.3% responded that accessibility is the responsibility of another team in the organization, among other answers with lower percentages (<2%) such as lack of time, lack of training and working on systems without interactive interfaces.

7 DISCUSSION

The searches carried out in the scientific databases ACM and IEEE did not return works that used methods and tools based on artificial intelligence concepts to validate accessibility requirements for people with disabilities. Considering this, when analyzing the results obtained by the SLM, we noticed the concentration of studies focused on traditional tools, methods and processes for the validation of accessibility requirements and, therefore, demonstrates the lack of application of artificial intelligence concepts in approaches that support development teams.
in validating accessibility requirements and improving access for people with disabilities.

Another inference made through the SLM is the lack of tools that support developers in assessing accessibility on different platforms beyond the web, such as mobile, desktop, and IoT applications. However, the processes and methods mentioned by the selected studies are adaptable to the development of accessible applications for different platforms and, thus, eliminate barriers for the interaction of different people with disabilities beyond the web.

Still on the MSL, the tools cited by the studies are more suitable for evaluating user interfaces. However, methods and processes are more suitable for measuring user experience, and meanings passed through software interaction, as they are cognitively evaluated by development professionals or accessibility specialists in one or more stages of the software lifecycle.

The mapping of tools indicated the existence of the “Web Accessibility Evaluation Tools List” database, which should be disseminated in the academic and organizational environments and used in the development of software projects. The base provides software developers with filtering options according to accessibility guidelines demanded in the software development project and presents a set of inspection tools to validate accessibility. However, another point
of attention is the lack of updated tools, as well as the number of tools that were released and are not maintained by the developers anymore.

Analyzing the volunteers’ responses to the questions in the questionnaire applied, the lack of concern with accessibility in the Brazilian software development scenario is evident. Although almost all volunteers (95.5%) deemed it important to consider accessibility in software development, only half indicated that they had already worked directly with accessibility. We can see a significant improvement in the percentage of developers who have worked in accessibility now versus in the first-ever research in this subject in Brazil (Freire et al., 2008), carried out in 2007, where only 19.9% of developers indicated having previous experience developing accessible software. However, only a slight increase in percentage comparing with another similar research carried out in 2018 (Antonelli et al., 2018) where 48.3% of the volunteers answered having already created accessible systems. Another improvement compared with the survey carried out by (Freire et al., 2008) is the percentage of organizations with accessibility concerns; while their research showed 35.7% of organizations in which the volunteers worked had accessibility concerns in some or all projects, we found that now, about 44% of the volunteer’s organizations have said concerns.

The responses also highlighted that few projects are concerned with these aspects in the context of their organizations, with only 43.2% of the volunteers’ responses indicating that they had already worked on projects that considered accessibility, even if indirectly (in a project with accessibility concerns but not working themselves with accessibility). According to the volunteers, accessibility is usually not a requirement neither for the organization (47%) nor for the client (68.2%), organizations which, for the most part (66.7%), do not have in their staff an accessibility specialist. In the survey carried out by (Freire et al., 2008) similar results were found, indicating that accessibility was not a requirement for the organization (51.3%) neither the clients (49.05%); this makes evident that accessibility was not and still not a priority in the software development process.

The WCAG was the accessibility guideline most indicated by the participants in the questionnaire and most cited in relation to the third SLM research question, which shows its importance both for the scientific community and the software development industry. According to the results of the mapping of accessibility tools, we could not find a tool that encompasses all accessibility variants according to WCAG rules.

Relating the results obtained in sections 4 (SLM) and 5 (Mapping of Available Accessibil-
ity Tools), we noticed that the work by Lim et al. (Lim et al., 2020) emphasized the use of the Axe Accessibility testing engine in Singapore’s government digital services; however, analyzing the answers to the questionnaire in section 6, it is clear that the volunteers did not indicate the use of this tool, for evaluating accessible software development. Therefore, it is clear that the tool is used in scientific works, but not by Brazilian developers who answered the questionnaire.

Finally, this study provided a general understanding of the accessibility scenario in academia and industry and the use of this fundamental concept for the development of inclusive applications. We also noticed that the union of the three investigation methods (MSL, tool mapping, and survey) contribute to the union and new challenges to be carried out by several professionals, such as researchers, designers, and developers, in favor of accessibility and social inclusion and digital of people, giving them more rights of access to information and communication.

Understanding how accessibility is considered in the development cycle has been a subject studied since 2006 (Tangarife & Mont’alvao, 2006). However, the results obtained through this research and Paiva et al. (2021) show that there has been much progress; however, there are still investigations to be carried out to support developers in the construction of accessible technologies.

8 FINAL CONSIDERATIONS AND FUTURE WORK

This work presented three ways to seek knowledge about accessibility validation approaches in software: an SLM, a mapping of accessibility tools, and a questionnaire applied to software development professionals. As a result of the MSL, we noticed the lack of intelligent approaches for validating accessibility requirements. Besides, many accessibility validation tools are outdated or disabled. On the questionnaire, we identified that the participants consider accessibility criteria an important factor in software development; however, few projects add it as a requirement in practice.

As future works, we intend to collect primary studies in other research bases (Scopus and Science Direct) to verify if other scientific repositories do not present research on intelligent approaches to validate accessibility in software. If the evidence remains the same as in this article, we intend to perform a new SLM, aiming to understand intelligent approaches that help software developers in validating requirements in general and, then, verify which of these methods and techniques are adaptable to the requirements of accessibility.

REFERENCES


³https://www.deque.com/axe/


Chandrasekaran, J., Ghandehari, L. S., Lei, Y., Kacker, R., & Kuhn, D. R. (2016). Evaluating the effectiveness of ben in localizing different types of software fault. In 2016 ieee ninth international conference on software testing, verification and validation workshops (icstw) (pp. 26–34).


