



Accessibility and Software Engineering Processes: A Systematic Literature Review

Débora Maria Barroso Paiva ^{a,*}, André Pimenta Freire ^b, Renata Pontin de Mattos Fortes ^c

^a Faculdade de Computação, Universidade Federal de Mato Grosso do Sul, Cidade Universitária, s/n - Campo Grande MS, Brazil

^b Departamento de Ciência da Computação, Universidade Federal de Lavras, Caixa Postal 3037 - Lavras MG, Brazil

^c Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, Av. Trabalhador São-carlense 400 - São Carlos SP, Brazil

ARTICLE INFO

Article history:

Received 3 March 2020

Received in revised form 30 July 2020

Accepted 10 September 2020

Available online 12 September 2020

Keywords:

Accessibility

Software Engineering

Systematic Literature Review

Design for disabilities

Methods for accessibility

ABSTRACT

Guidelines, techniques, and methods have been presented in the literature in recent years to contribute to the development of accessible software and to promote digital inclusion. Considering that software product quality depends on the quality of the development process, researchers have investigated how to include accessibility during the software development process in order to obtain accessible software. Two Systematic Literature Reviews (SLR) have been conducted in the past to identify such research initiatives. This paper presents a new SLR, considering the period from 2011 to 2019. The review of 94 primary studies showed the distribution of publications on different phases of the software life cycle, mainly the design and testing phases. The study also identified, for the first time, papers about accessibility and software process establishment. This result reinforces that, in fact, accessibility is not characterized as a property of the final software only. Instead, it evolves over the software life cycle. Besides, this study aims to provide designers and developers with an updated view of methods, tools, and other assets that contribute to process enrichment, valuing accessibility, as well as shows the gaps and challenges which deserve to be investigated.

© 2020 Elsevier Inc. All rights reserved.

1. Introduction

The expansion of the web, the presentation of content in different formats and devices (e.g., audio, video, mobile phones, tablets) and the emergence of new platforms (e.g., Internet of Things) have motivated the conduction of several research studies about digital accessibility in recent years. According to W3C's (World Wide Web Consortium) WAI (Web Accessibility Initiative), accessibility means that people with disabilities, reduced skills, or situational induced impairments should be able to access, navigate, interact, and contribute to information available on computers, electronic equipment, and the Internet (World Wide Web Consortium, 2019a).

In particular, the importance of digital accessibility has been reinforced, as digital media have become the primary source for citizens to access to government, educational, news, and leisure information. The W3C launched an initiative whose main mission is to coordinate international, technical, and human efforts to improve web accessibility (World Wide Web Consortium, 2019b). The WAI initiative is responsible for the important set of accessibility guidelines, called Web Content Accessibility Guidelines (WCAG) (World Wide Web Consortium, 2019c).

An extensive body of research has long recognized that software design must address the diverse needs of users represented across the population, including older and disabled people. The discussions, extensively presented in the literature, focused on proposing, disseminating, and evaluating guidelines, that addressed the design of accessible software (Gulliksen et al., 2004; Wentzel et al., 2016b; Wille et al., 2016). Besides, more recently, researchers have highlighted the importance of considering accessibility throughout the software life cycle (Krainz et al., 2018; Reichling and Cherfi, 2013; Horton and Sloan, 2014).

Freire et al. (2007b) and Dias et al. (2010) analyzed the insertion of accessibility in the context of Software Engineering processes through Systematic Literature Reviews (SLR). The first one regarded the period from 1998 to 2006, and the second one regarded the period from 2007 to 2010. This article has a primary objective to present results on the updating of the previous SLR, now focusing on a more recent period, the works from 2011 to 2019, but considering a similar protocol. In general, SLR is a means of aggregating knowledge about a topic or research question. The SLR methodology aims to be as unbiased as possible by being auditable and repeatable (Kitchenham et al., 2010).

This paper is organized as follows: Section 2 presents concepts about accessibility and related work; Section 3 describes the research method used; Section 4 presents the results of the SLR; Section 5 discusses the main findings of the study and their implications; and Section 6 presents final remarks of the paper.

* Corresponding author.

E-mail addresses: debora.paiva@ufms.br (D.M.B. Paiva), apfreire@dcc.ufma.br (A.P. Freire), renata@icmc.usp.br (R.P. de Mattos Fortes).

2. Background and related literature reviews

Several factors can influence the implementation of digital accessibility, such as the knowledge of designers and developers. Developers usually justify on how they deal with accessibility assurance in their projects, by reporting that even the size of an organization and the different resources such as time and costs involved can influence. Different methods, standards, and guidelines were proposed in the last years aiming to describe critical points that should be considered by stakeholders.

One of the most widely used sets of accessibility recommendations is the WCAG. The WCAG, currently in version 2.1 published in 2018, is a document that explains, through guidelines and recommendations, how to make web content accessible to people with disabilities (World Wide Web Consortium, 2019c). WCAG aims to help front-end developers, but it is also useful for assisting developers of assessment tools, developers of audit tools, and developers of quality assurance and validation tools. It uses A, AA, and AAA categories to classify a Web content in different levels, based on its conformance with WCAG standards. By following WCAG 2.1 document, developers will be able to make content more accessible to a broader range of people with disabilities, including blindness and low vision, deafness and hearing loss, limited movement, speech disabilities, photo-sensitivity, and also guidelines for cognitive limitations and learning disabilities.

Software Engineering plays a fundamental role in the development of accessible applications since it can promote the integration between methodologies and specific accessibility techniques and activities at a software development process. Many studies have been proposed aiming to integrate accessibility with software processes (Sanchez-Gordon et al., 2019; Bouraoui and Gharbi, 2019; Sánchez-Gordón and Moreno, 2014).

Freire et al. (2007a) carried out an SLR to identify methods, techniques, and approaches for the treatment of web accessibility, from Web Engineering's point of view. The authors obtained a panorama of the current techniques and classified them according to their adequacy in the different phases of the Web Engineering process. The research question was: "What are the current methods, techniques, processes, and approaches in the literature to support the development of accessible web applications?"

Their results included an analysis of 53 primary studies, selected from an initial set of 844. The authors classified the found studies as a support technique for one or more processes of ISO/IEC 12207 (ISO/IEC, 1998). They found studies on requirements engineering (1 study), architectural design (5 studies), navigational design (2), user interface design (14), content design (8), software development (3), software testing (31), software maintenance (5), software measurement (7), and training (3). The authors observed a growing interest in the subject during the period. They also pointed out several research gaps since, in some of the ISO/IEC 12207 areas, no work had been identified at the time.

Dias et al. (2010) updated Freire et al.'s research considering the period from 2007 to 2010. The results included an analysis of 65 studies selected from an initial set of 301. The results indicated that there were studies in the areas of requirements engineering (13), architectural design (3), navigational design (10), user interface design (14), content design (7), software development (7), software testing (33), software maintenance (4), and software integration (2). From the analyzed data, they concluded that research on web accessibility increased during the period. Although there were many accessibility techniques in software development, the study also indicated some activities related to Web Engineering had sparse techniques for accessible applications.

3. Research method

We considered guidelines provided by Kitchenham and Charters (2007) and Petersen et al. (2015). In general, the main activities suggested by these authors are (i) planning, (ii) conducting, and (iii) reporting the review. These activities are detailed in the next subsections regarding our SLR.

3.1. Planning the review

The main sub-activities related to planning the SLR are the specification of the research questions, the search string, and the search strategy. For this SLR, our research questions are:

- **Research Question 1 (RQ1):** "What are the current methods, techniques, processes, and approaches to support the development of accessible software?"
- **Research Question 2 (RQ2):** "What are the main assets (methods, techniques, tools, etc.) available to contribute to the development of accessible software?"

To define the search string, we selected a set of keywords from studies about Software Engineering and accessibility and matched them with our research questions. We also established synonyms and alternative spellings and optimized the keywords iteratively. We tested the set of keywords in different databases and refined it. Using Boolean operators to connect the keywords, we defined the following search string:

(Software Engineering and accessibility) or (methods and accessibility) or (techniques and accessibility) or (process and accessibility) or (frameworks and accessibility).

Our search strategy included the following procedures: (i) searching and extracting the studies using the search string, and published from 2011 to 2019, gathered from different databases; (ii) eliminating the duplicate studies; (iii) filtering publications by title, keywords, and abstract analysis using inclusion and exclusion criteria; (iv) reading the selected papers; (v) extracting data (use of an extraction form); and (vi) synthesizing the results.

Sources list included several journals and conference proceedings available on the web digital libraries: those ones indexed by IEEE Xplore,¹ ACM Digital Library,² Springerlink,³ Science Direct,⁴ and Wiley Interscience.⁵ For inclusion criteria, we considered papers available on the web in electronic format, written in English or Portuguese, and describing techniques for accessibility for any phase of the development of web software process or management activities or process improvement activities or software quality activities. The search string was adapted for every digital library. Exclusion criteria covered papers describing exploratory studies (for example, teaching accessibility in the classroom), poster or abstract, papers not accessible free of charge, and development of assistive technologies because they are not directly related to techniques to be used by designers and developers. One of the researchers developed the review protocol that was revised by another one, also involved in this study.

¹ <http://ieeexplore.ieee.org>.

² <http://portal.acm.org>.

³ <http://www.springerlink.com>.

⁴ <http://www.sciencedirect.com>.

⁵ <http://interscience.wiley.com>.

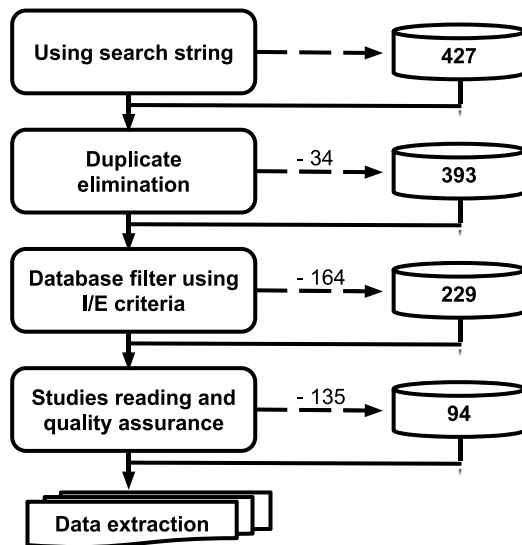


Fig. 1. Search strategy used in this SLR.

3.2. Conducting the review

The researchers carried out searches in the sources during 16–21 September 2019, returning 427 papers. Firstly, duplicates were eliminated, and inclusion and exclusion criteria were applied. Then we had gotten 229 papers, which were read (full text). Also, we applied the quality assurance criteria (presented in Section 3.3), and 94 papers were finally selected. Therefore, we could examine more details about each selected paper and extract data using a form. Fig. 1 presents an overview of our search strategy.

3.3. Data extraction and analysis

Initially, we extracted the main process areas in ISO/IEC/IEEE 12207 Standard (ISO/IEC/IEEE, 2017) and created our analysis categories. From the complete reading of the articles, it was possible to identify if the papers were related to such analysis categories and to assort them. The main technical terms repeated in the article (keywords) helped in the categorization. For example, the words requirements elicitation, functional and non-functional requirements suggested the “Requirements” process area.

Concerning the quality assurance, we have based on evaluation concepts discussed by Ivarsson and Gorschek (2011). We considered the following questions to evaluated the quality of the 94 selected papers:

1. Can the context of development and evaluation of the paper be understood? Does it involve, for example, a description of the academic or industrial environment and the considered disability?
2. Are the design and implementation of the proposal described? For example, does it involve indicating programming language, modeling, and database tools used and the created diagrams?
3. Does the paper propose at least one asset?
4. Does the paper present any evaluation of the proposal?

In the course of studies reading and quality assurance step (Fig. 1), we considered to include only papers with definite answers to at least three quality assurance questions. One author conducted the review and another one supervised all activities. We defined an extraction form and spreadsheet to support the identification of individual elements of each selected paper. The form is presented in Table 1.

Table 1

Information extracted in the form from the selected papers.

| Information | Options |
|---|-----------------|
| 1. Paper ID | |
| 2. Title | |
| 3. Abstract | |
| 4. Software development phase or activity discussed | |
| 5. Asset presented | Yes (which?)/no |
| 6. Context description | Yes (which?)/no |
| 7. Design and implementation description | Yes (which?)/no |
| 8. Evaluation description | Yes (which?)/no |
| 9. Local and data | |

4. Results

In general, the selected papers were related to web, mobile, and IoT platforms. Fig. 2 presents the accounted results. We could observe that researches are concentrated mainly on the technical processes, especially the design and the testing processes. Fourteen studies emphasized two activities, which are 7 (R&D) + 3 (D&I) + 2 (R&T) + 1 (D&Ma) + 1 (T&Me), as presented in the Venn Diagram, Fig. 3.

All activities indicated in previous SLRs have been observed in the current work. The exception was the integration activity, which occurred in two papers analyzed by Dias et al. but did not occur in the current study. Additionally, new activities were observed in this study: process establishment (9 papers) and process improvement (1 paper).

As highlighted by Nganji and Nggada, software engineers need to consider the range of disabilities present in people involved with the aimed software, such as visual and hearing impairments and adapt existing processes (Nganji and Nggada, 2011). Similarly, the concern was observed in a study by Horton and Sloan (2014). The authors point out that accessibility audits are typically performed during the phases of quality assurance, and user acceptance testing and solutions for the issues identified usually take place in code. The best fix for many complex accessibility issues would be to revisit the overall approach.

Such studies also reveal the recent interest in improving software life cycles, including accessibility, intending to obtain accessible software. In fact, in the past, studies have indicated that product quality depends on the process quality (Humphrey, 1989; Fuggetta, 2000). We can now realize that accessibility issues have begun to be incorporated into software life cycles, creating a new research subject and contributing to the solution of many problems related to software accessibility.

Following, we describe the analysis for classification of the studies of this SLR into the software process phases, as well as the main assets the studies have promoted, keeping in mind the two research questions.

4.1. RQ1: What are the current methods, techniques, processes, and approaches to support the development of accessible software?

By looking for the answers to this first question, in the 94 selected papers, we observed they were related to approaches that could be grouped in nine software process phases and support activities (software requirements, design, implementation, testing, maintenance, process establishment, training, measurement, and process improvement), as detailed in the next subsections.

4.1.1. Software requirements

Twenty-three papers (representing 24.4% of the total) were related to software requirements. These studies could be grouped into three main topics of interest, as presented in Table 2.

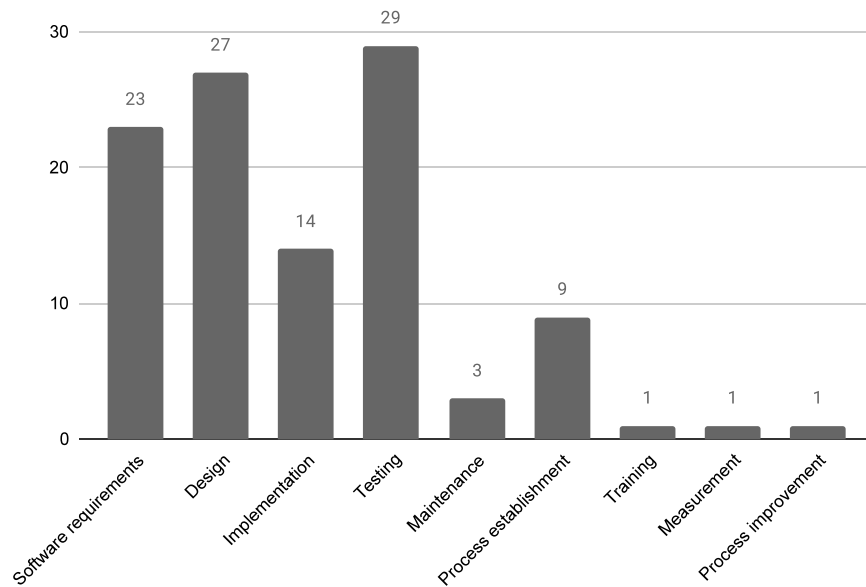


Fig. 2. Number of papers considering each software process phases or activities, articles related to any support to developing accessible software.

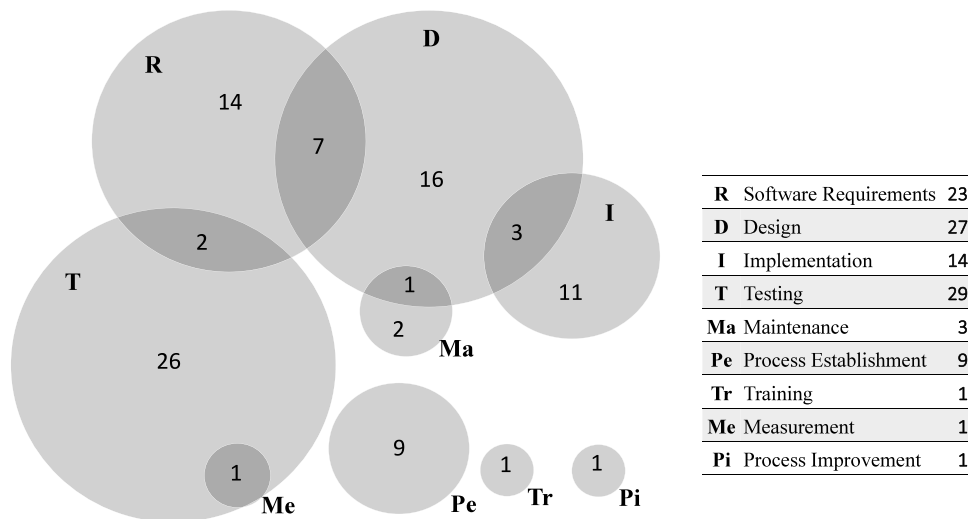


Fig. 3. Venn Diagram showing the number of papers found that refers to certain(s) software process phase(s) or activity(ies).

New techniques for eliciting accessibility requirements were presented. [Shirogane \(2014\)](#) mentions that there are many accessibility guidelines, but applying them to all elements of the software is expensive. The author has proposed a method to elicit accessibility requirements in the early software development phase. It is based on checklists to analyze the environment and the problems of end-users, obtaining the accessibility requirements as a priority.

[Sirikitsathian et al. \(2017\)](#) discuss the lack of knowledge and understanding of the principles that affect universal web accessibility by designers and developers, and they proposed a conceptual framework of web accessibility and acceptance by users with visual impairment. They highlighted that the framework should be useful for designers, developers, and related personnel with developing websites, web browsers, web tools, and applications so that they will recognize and understand the accessibility requirements associated with the problems faced by visually impaired users.

Much interesting research could be obtained on techniques for requirements elicitation for dealing with people with different types of disabilities and for the elderly. In this way, an ontology was proposed to model the knowledge of accessibility guidelines

related to learning objects ([Ingavélez-Guerra et al., 2018](#)). Similarly, [Cairns et al. \(2019\)](#) propose developing vocabulary and language for game accessibility to ensure that disabled people have the experience they want to have. The vocabulary consists of two broad types of options in access and challenge, which focus on the player-loop and the game world. Authors focused on nine different categories related to access options (input, control, presentation, and output) and challenge options (performance, training, progress, social, and moderation).

Collections of accessibility requirements were also presented in the literature. [Schulz and Fritsch \(2014\)](#) analyzed the needed requirements for making the e-identity infrastructure (a combination of username and password) accessible and usable by the most people possible. They found requirements in how to present information, how the interface needs to be controlled and tested, and how to provide help and support.

[Peissner et al. \(2014\)](#) present a collection of requirements for systems with adaptive user interfaces that can enable widespread market adoption. [Ferati et al. \(2012\)](#) suggest that the one-solution-fits-all model is inadequate because it does not consider the levels of visual impairment when providing customized web

Table 2

The twenty-three selected papers which presented approaches related to *Software requirements (R)*, grouped by the main topics of interest.

| Main groups of topics of interest | References |
|--|--------------------------------|
| R1. Accessibility requirements definition | Dias et al. (2012) |
| | Ferati et al. (2012) |
| | Ferreira and Bonacin (2014) |
| | Peissner et al. (2014) |
| | Schulz and Fritsch (2014) |
| R2. Engineering approaches to include accessibility issues | Miñón et al. (2014) |
| | Shirogane (2014) |
| | Ludi (2012) |
| | Johari and Kaur (2013) |
| | Chalkia and Bekiaris (2011) |
| | Pellegrini et al. (2019) |
| | Ferreira et al. (2019) |
| | Cairns et al. (2019) |
| | Ludi et al. (2012) |
| | Ingavélez-Guerra et al. (2018) |
| | Sirikitsathian et al. (2017) |
| | Engiel et al. (2014) |
| | Oliveira et al. (2016) |
| | Ferati et al. (2016) |
| | Ross et al. (2017) |
| | Heron et al. (2013a) |
| | Liu et al. (2018) |
| R3. Tool for gathering accessibility requirements | Chiplunkar et al. (2019) |

experience. They proposed a set of requirements to guide the process of building a middleware prototype. The prototype built using eight adaptation techniques provides the required user experience based on a user's level of visual impairment.

4.1.2. Design

Twenty-seven papers (representing 28.7% of the total) were related to software design. These studies could be grouped into nine main topics of interest, as presented in Table 3.

Wentzel et al. (2016b) specified guidelines that enable developers to design accessible wearable apps. Their guidelines were created by developers who have experience with designing accessible technology or wearables. Besides, for developing the guidelines, users who experience problems with the accessibility of wearables (persons with some disability) ensured their validity from an end-user perspective.

Peissner et al. (2012) present a generic infrastructure, MyUI, to increase accessibility through adaptive user interfaces automatically generated. The multimodal design patterns repository serves as the basis for a modular approach to individualized user interfaces. Raufi et al. (2015) also investigated adaptive user interfaces and provided an overview of a survey concerning adaptive interaction between users and web information space. The emphasis was on methods and techniques for adaptive web accessibility for blind and visually impaired people. Also, they introduced various visual and auditory approaches to extend such methods and techniques of adaptive products for blind and visually impaired people.

Both scientific studies of Junior and Piazza (2017) and Angkananon et al. (2015) investigated design definition considering accessibility for specific contexts. The first one presents a component framework developed from the viewpoint of accessibility, based on a pattern language focusing on e-commerce and the accessibility guidelines targeted to blind people. The second one describes a Technology Enhanced Interaction Framework (TEIF) as well as a method that has been shown having the potential to help developers make local Thai museums more accessible for disabled visitors (hearing impairment).

Table 3

The twenty-seven selected papers which presented approaches related to *Software design (D)*, grouped by the main topics of interest.

| Main groups of topics of interest | References |
|--|--------------------------------|
| D1. Accessible wearable app design | Wentzel et al. (2016b) |
| | Liu et al. (2018) |
| | Alabi and Gooch (2011) |
| | Wentzel et al. (2016a) |
| D2. Adaptive user interfaces | Peissner et al. (2012) |
| | Raufi et al. (2015) |
| | Engel et al. (2019) |
| | Oliveira et al. (2018) |
| D3. Accessibility for specific contexts | Junior and Piazza (2017) |
| | Angkananon et al. (2015) |
| D4. Design for disabilities | Ferati et al. (2012) |
| | Johari and Kaur (2013) |
| | Scott et al. (2015) |
| | Rothberg (2019) |
| D5. Ontology, knowledge modeling | Ingavélez-Guerra et al. (2018) |
| D6. New representation (design) of web pages | Idrobo et al. (2017) |
| | Semaan et al. (2013) |
| D7. Architecture for accessible web pages | Chalkia and Bekiaris (2011) |
| | Miñón et al. (2014) |
| | Calle-Jimenez et al. (2018) |
| | Martín et al. (2011) |
| | Huang et al. (2015) |
| D8. Design patterns | Sanchez-Gordon et al. (2019) |
| | Fogli et al. (2014) |
| | Zaki and Forbrig (2011) |
| | Engiel et al. (2014) |
| D9. Model driven engineering | Bouraoui and Gharbi (2019) |

The VERITAS framework, proposed by Scott et al. (2015), provides designers with the capability of choosing designs for attending a wide range of disabilities defined within the VERITAS repository and generating a virtual user model based on particular disabilities. The goal of the VERITAS framework is to provide support to designers as they evaluate the accessibility of their designs. To this end, the framework generates a report which highlights crucial problems and presents relevant statistics based on a simulation.

4.1.3. Implementation

Fourteen papers (representing 14.8% of the total) were related to software implementation. These studies could be grouped into five main topics of interest, as presented in Table 4.

Many research studies proposed frameworks. Jain and Singh (2011) discuss how speech can be effectively integrated into the interfaces and also introduce a framework for programming languages that would allow the rapid development of such interfaces. Alayed et al. (2016) presented a framework for the development of localized web accessibility guidelines to enhance the accessibility of Saudi university websites considering issues related to culture and language. Oliveira et al. (2014) proposed the Homero framework in order to support the development of an accessible interface layer of web applications. Developed using the PHP language, it automates the generation of web pages following guidelines defined in WCAG 2.0. Rodriguez-Ascaso et al. (2011) developed a general framework implemented through a standard-based open architecture focused on supporting inclusive learning at Higher Education. The services involve the stakeholders (students and professionals) to attend users' needs; services are provided through standards-based framework components, which combine user-centered design tools with modeling techniques based on the state-of-the-art on semantic web technologies.

Table 4

The fourteen selected papers which presented approaches related to *Software implementation (I)*, grouped by the main topics of interest.

| Main groups of topics of interest | References |
|---|---|
| I1. Techniques to implement accessible products | Crabb et al. (2019) Jain and Singh (2011) Ferati and Sulejmani (2016) Salah et al. (2017) |
| I2. Adaptive user interfaces implementation | Peissner et al. (2012) |
| I3. Library of software code | Shirogane et al. (2011) Basman et al. (2011) |
| I4. Accessibility and standards | Idrobo et al. (2017) |
| I5. Framework to build accessible software | Huang et al. (2015) Kaklanis et al. (2011) Alayed et al. (2016) Rodríguez-Ascaso et al. (2011) Heron et al. (2013b) Oliveira et al. (2014) |

Table 5

The twenty-nine selected papers which presented approaches related to *Software testing (T)*, grouped by the main topics of interest.

| Main groups of topics of interest | References |
|---|--|
| T1. Evaluation tools | Chiplunkar et al. (2019) Vigo et al. (2013) Oikonomou et al. (2011a) Sohaib et al. (2018) Hilera et al. (2018) Avila et al. (2012a) Dongaonkar et al. (2017) |
| T2. Automatic detection of accessibility problems | Crespo et al. (2016) Moura et al. (2017) Snaprud et al. (2014) |
| T3. Mobile apps testing | Eler et al. (2018) Patil et al. (2016) |
| T4. Framework for accessibility evaluation/tests | Tuan et al. (2012) Acosta-Vargas et al. (2018) Rodríguez et al. (2017) Oikonomou et al. (2011b) Fernandes et al. (2011) |
| T5. Methods for accessibility evaluation | Sánchez-Gordón and Moreno (2014) Ferreira et al. (2019) Acosta-Vargas et al. (2019a) Avila et al. (2012b) Acosta-Vargas et al. (2019b) Watanabe et al. (2012) Malý et al. (2012) Giakoumis et al. (2014) Zhang et al. (2015) Watanabe et al. (2017) Orozco et al. (2016) |
| T6. Evaluation/tests and standards | Wille et al. (2016) |

Additionally, Ferati and Sulejmani (2016) present a system that automatically increases web page accessibility by applying three different techniques: link, image, and navigation enrichment.

4.1.4. Testing

Twenty-nine papers (representing 30.8% of the total) were related to software testing. These studies could be grouped into six main topics of interest, as presented in Table 5.

Oikonomou et al. (2011a) introduced a personalized web accessibility evaluation methodology, allowing for web content accessibility evaluation regarding different disability profiles (disabilities, personas) as well as resources of assistive technologies and devices.

Regarding mobile tests, Eler et al. (2018) introduced the idea of using automated test generation to explore the accessibility of mobile apps. They presented the MATE tool (Mobile Accessibility Testing), which automatically verifies apps by performing different checks for accessibility issues related to visual impairment. Moura et al. (2017) provided an API that automatically runs tests, analyzing whether the interfaces of an application conform to accessibility rules and offer tips for improving accessibility. Patil et al. (2016) presented a tool for evaluation of application accessibility in the Android environment. It intends to identify the best color combination for foreground and background to make Android applications accessible for users with low vision and color blindness.

Watanabe et al. (2012) described an approach for testing accessibility requirements in Rich Internet Applications (RIA), by using acceptance tests. The authors had implemented a set of assistive technology user scenarios in the acceptance tests in order to guarantee keyboard accessibility in web applications. As the scenarios were implemented as acceptance test scenarios, they provided accessibility analysis over all layers of the software, from server-side to client-side implementations in RIA.

Acosta-Vargas et al. (2019a) argued that there are no adequate tools and methods for evaluating the accessibility of multimedia resources, which makes the type of video resource not accessible to all users. They suggest applying a combined method between the automatic tool for photosensitive epilepsy and manual evaluation with the "Website Accessibility Conformance Evaluation Methodology". This study contributes to improving the accessibility of multimedia resources.

Avila et al. (2012b) introduced a tool to support accessibility evaluation in the generated web content. Using the tool users can identify web accessibility failures that can be improved. The authors also introduced a new approach for the web-content manual accessibility evaluation process by applying the Case-Based Reasoning Technique.

Considering accessibility evaluation methods, Zhang et al. (2015) mentioned that all the pages in a website need to be analyzed for clustering, causing huge computation costs. To address this issue, they proposed a novel page sampling method based on URL clustering for web accessibility evaluation. Using only the URL information for stratified page sampling, the tool can efficiently scale to large websites. By exploiting similarities in URL patterns, the tool cluster pages and can effectively detect accessibility problems from web page templates.

Fernandes et al. (2011) considered that the web pages presented to users differ from the original structure and content transmitted through HTTP communication. Then, they considered that this poses a series of challenges for Web accessibility evaluation, especially on automated evaluation software. The authors proposed an evaluation framework for performing web accessibility evaluations in different environments to understand how similar or distinct these environments can be in terms of their web accessibility quality.

Giakoumis et al. (2014) presented web-service accessibility assessment techniques through a unified Quality of Services (QoS) context. The goal was to enable QoS-aware service selection systems to select and provide accessible web services, ones that are properly designed to allow their consumption from end-user applications. Therefore, a web-service accessibility assessment framework (WSaaS) was developed, based on accessibility guidelines, dealing with accessibility issues that can appear both on

Table 6

The three selected papers which presented approaches related to *Software maintenance (Ma)*, grouped by the main topics of interest.

| Main groups of topics of interest | References |
|-----------------------------------|---|
| Ma1. Refactoring techniques | Garrido et al. (2012) Ikhsan and Candra (2018) |
| Ma2. Reengineering | Oliveira et al. (2018) |

Table 7

The nine selected papers which presented approaches related to *Process establishment (Pe)*, grouped by the main topics of interest.

| Main groups of topics of interest | References |
|---|---|
| Pe1. Accessibility and Software Engineering | Reichling and Cherfi (2013) Horton and Sloan (2014) Nganji and Nggada (2011) Silva et al. (2017) Andrade et al. (2018) Conway and Fitzpatrick (2017) Velleman et al. (2017) |
| Pe2. Case tool for accessibility and Software Engineering | Branco et al. (2014) |
| Pe3. Accessible software processes | García-Borgoñón et al. (2014) |

the presentation level of content delivered through web services and on the content level itself. [Crespo et al. \(2016\)](#) proposed the Social4all, a platform able to analyze websites and detect accessibility problems automatically; after this, a guided assistant is used to offer adequate solutions to each recognized problem.

4.1.5. Maintenance

Three papers (representing 3.2% of the total) were related to software maintenance. These studies could be grouped into two main topics of interest, as presented in [Table 6](#).

[Garrido et al. \(2012\)](#) have proposed empowering users with the ability to select, in their client browsers, their interface refactoring for each site they access. The approach is called Client-Side Web Refactoring. It allows for the automatic creation of different, personalized views of the same application to solve the particular bad smells that each user recognizes.

More recently, [Ikhsan and Candra \(2018\)](#) developed a method and a tool to improve web accessibility automatically (automated refactoring), aiming to refactor faster than manually. The Automatically tool derived from modifying an existing automated evaluation tool, namely HTML Code Sniffer, and implementing 40 techniques from the WCAG that can be automated.

[Oliveira et al. \(2018\)](#) present a strategy to promote web accessibility in legacy systems through the use of software awareness. They use self-adaptation strategies to provide customized accessibility and discuss an application in reengineering a real system based on Android.

4.1.6. Process establishment

Nine papers (representing 9.5% of the total) were related to process-establishment. These studies could be grouped into three main topics of interest, as presented in [Table 7](#).

[Andrade et al. \(2018\)](#) presented the Acero approach, which integrates accessibility into the software development process. Also, the authors presented the Acero tool, which provides the computational support to allow the automation of the Acero approach. The inclusion of accessibility is transparent, i.e., when using the Acero tool, professionals will be able to generate accessible applications without being experts in the area.

[Velleman et al. \(2017\)](#) identified key factors that influence the adoption and implementation of accessibility standards for

Table 8

The three selected papers which presented approaches related to *Training (Tr)*, *Measurement (Me)*, and *Process improvement (Pi)*, grouped by the topics of interest.

| Main topics of interest | References |
|---|---|
| Tr1. Gamification for accessibility training | Spyridonis et al. (2017) |
| Me1. Measure the fulfillment of the WCAG 2.0 | Wille et al. (2016) |
| Pi1. Capability maturity of usability and accessibility | Quintal and Macías (2018) |

local government websites and place them in a new adoption and implementation model, using a process-oriented approach. The model includes many factors, such as budget, costs, knowledge, and quality assurance.

[Nganji and Nggada \(2011\)](#) reinforce that while traditional life cycles ensure that the system meets the needs of people without disabilities, the needs of people with disabilities are often overlooked thus resulting in systems that are inaccessible and unusable to them. Authors proposed a disability-aware software engineering process model which considers the needs of people with disabilities, hence improving accessibility and usability of the designed system. The process model contains twelve phases, including establishing the need for the system, creating personas, acquiring identified technologies, design, implementation, test, evaluation, and maintenance.

Additionally, [Horton and Sloan \(2014\)](#) indicated that the best fix for many complex accessibility issues might be to revisit the design approach, yet reworking designs at this late phase has a significant impact on the project.

4.1.7. Training, measurement, and process improvement

One paper was related to training, one paper was related to measurement, and one paper was related to process improvement (representing 3.2% of the total), as presented in [Table 8](#).

Original research on accessibility training was developed by [Spyridonis et al. \(2017\)](#). They proposed a framework using gamification as a method for engaging and motivating web designers to increase the adoption of the WCAG 2.0 guidelines. The authors extended the gamification concepts by providing a framework that maps real-world elements of the task under study to elements of games for improving awareness and increasing adoption of the available web accessibility guidelines. For example, considering the Alternative Text guideline, i.e., all non-text content presented to users should have a text alternative. This guideline has been mapped to a "Mystery Box" mechanic, where users are in a dark room viewing an unclear picture. The Assistive Robot will explain what is on the image and the player must write the description in the 'ALT' tag of the image to progress.

Concerning the measurement process, [Wille et al. \(2016\)](#) presented an approach to measure and calculate the fulfillment of the WCAG 2.0. Authors based on an empirical study and evaluation of existing automatic test tools. The fulfillment value serves for evaluating how good web content fulfills the success criteria. It also allows estimation about the effort to change an existing web content to fulfill the WCAG 2.0 guidelines. The approach is developed as a prototype, which initially evaluates parts of the success criteria of WCAG 2.0. This prototype supports human tester in order to evaluate web page accessibility in an automatic or semi-automatic manner. It is worth noting that only one study addressed measurement of accessibility in Software Engineering, while previous studies showed a considerable number of studies in the area in the years between 2000 and 2010 ([Freire et al., 2007a](#); [Vigo and Brajnik, 2011](#)).

Table 9

The distribution of all the groups of the topics of interest evidenced by the selected papers, related to the nine processes.

| Processes | Groups of the main topics of interest |
|----------------------------|---|
| Requirements (R) | R1. Accessibility requirements definition R2. Engineering approaches to include accessibility issues R3. Tool for gathering accessibility requirements |
| Design (D) | D1. Accessible wearable app design D2. Adaptive user interfaces D3. Accessibility for specific contexts D4. Design for disabilities D5. Ontology knowledge modeling D6. New representation (design) of web pages D7. Architecture for accessible web pages D8. Design patterns D9. Model driven engineering |
| Implementation (I) | I1. Techniques to implement accessible products I2. Adaptive user interfaces implementation I3. Library of software code I4. Accessibility and standards I5. Framework to build accessible software |
| Testing (T) | T1. Evaluation tools T2. Automatic detection of accessibility problems T3. Mobile apps testing T4. Framework for accessibility evaluation/tests T5. Methods to accessibility evaluation T6. Evaluation/tests and standards |
| Maintenance (Ma) | Ma1. Refactoring technique Ma2. Reengineering |
| Process establishment (Pe) | Pe1. Accessibility and Software Engineering Pe2. Case tool for accessibility and Software Engineering Pe3. Accessible software processes |
| Training (Tr) | Tr. Gamification for accessibility training |
| Measurement (Me) | Me. Measure the fulfillment of the WCAG 2.0 |
| Process improvement (Pi) | Pi. Capability maturity of usability and accessibility |

Quintal and Macías (2018) stated, on process improvement, that the capability maturity of usability and accessibility processes should be considered and measured. To take up this challenge, the authors described the conception of a capability maturity model named MODECUA, which provides a framework to evaluate development processes based on usability and accessibility. The proposed model was developed by extending and adapting the general capability maturity models.

4.2. RQ2: *What are the main assets (methods, tools, etc.) available to contribute to the development of accessible software?*

We have analyzed in the selected papers which assets are offered by researchers, as presented in Fig. 4. Methods and frameworks are the main results of researches in this area, demonstrating that the scientific community provides mechanisms that contribute to the implementation of accessibility in practice. Methods and processes are particularly important because they can be adapted and instantiated for different platforms, such as web and mobile, and many tools can be developed from them.

In the search for assets in the 94 selected papers, we had evidence of the whole effort in these researches and Table 9 summarizes the 31 groups of topics of interest related to the nine processes. Fig. 5 also indicates the number of selected papers. Section 4.3 briefly describes the assets presented in the selected papers according to the groups of topics of interest.

4.3. Assets description

The nine groups showed in Table 9 also include research regarding a variety of assets. Following, the addressed assets are listed and described according to the activity in the development process they support. We aimed to highlight the efforts to generate the technical apparatus related to the developers' and engineers' activities, supporting the process.

1. For Requirements (R)

R1. Assets for requirements elicitation: (1) *set of requirements* resulting from workshops with stakeholders addressing the needs of the visually impaired community (Ferati et al., 2012); (2) *method* to include accessibility requirements in a Web Engineering Framework (Dias et al., 2012); (3) *set of requirements* about how information must be presented, how the interface needs to be tested, and how to provide help and support considering accessibility (Schulz and Fritsch, 2014); (4) *set of requirements* for adaptive user interface systems that can enable widespread market adoption (Peissner et al., 2014); and (5) *set of requirements and recommendations* for the design of Web content, mobile, and assistive technology; *method* to be used in participatory practices with people with hearing loss (Ferreira and Bonacin, 2014).

Therefore, in Group R1. (requirements elicitation), we identified the following assets: 4 *set of requirements* and 2 *methods*.

R2. Assets for engineering approaches to include accessibility requirements: (1) *conceptual framework* to incorporate accessibility in mobile apps inspired by epidemiology (Ross et al., 2017) or acceptance models (Sirikitsathian et al., 2017); (2) *semi-automated method* for the elicitation of accessibility requirements and OmnesWeb tool to support the method (Oliveira et al., 2016); (3) *techniques* to help the team gather the needs and tasks of math/science instructors and visually impaired students (Ludi, 2012; Ludi et al., 2012); (4) *set of guidelines* to ensure accessibility in early phases of software development projects (Pellegrini et al., 2019); (5) *method* to elicit accessibility requirements in the early software development phase (Shirogane et al., 2011); (6) *vocabulary* of game accessibility focused on the experience of disabled users (Cairns et al., 2019); (7) *method* to perform requirements elicitation using legal resources and patterns (Engiel et al., 2014); (8) *ontology* to model accessibility requirements of learning objects (Ingavélez-Guerra et al., 2018); (9) *method* to integration of accessibility requirements into a user interface development process; SPA4USXML tool to support the method (Miñón et al., 2014); (10) *method* to create a new version of an application that had no accessibility at all (Ferreira et al., 2019); (11) *method* to incorporate accessibility requirements in mobile applications (Johari and Kaur, 2013); (12) *method* augmenting accessibility requirements gathering by indirect observation of disabled people using Internet of Things (IoT) devices (Ferati et al., 2016); (13) *method* to correlate and link the characteristics of the disabled users with their functional limitations, the assistive technologies and the design guidance (Chalkia and Bekiaris, 2011); (14) *technique* to motivate participation when building accessible maps mobile applications (Liu et al., 2018); and (15) *ACCESS software framework* aimed at addressing substantial problems associated with supporting novice users with special interaction needs (Heron et al., 2013a).

Therefore, in Group R2. (engineering approaches to include accessibility requirements), we identified the following assets: 2 *frameworks*, 8 *methods*, 2 *techniques*, 1 *set of guidelines*, 1 *vocabulary*, 1 *ontology*, and 1 *tool*.

R3. Asset for tool for gathering accessibility requirements: (1) *Drawxi tool* for free-form sketching and sharing simple diagrams (processes, workflows, ideas, perspectives, etc.) (Chiplunkar et al., 2019).

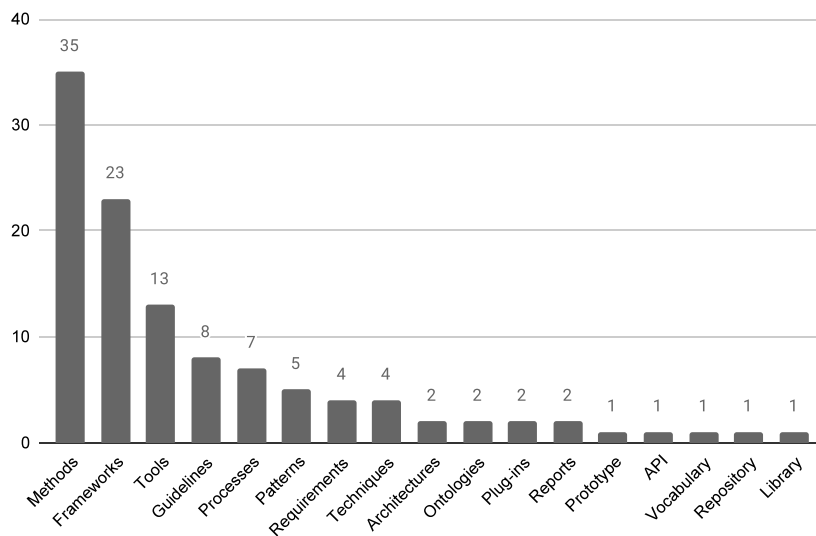


Fig. 4. Assets resulting from research.

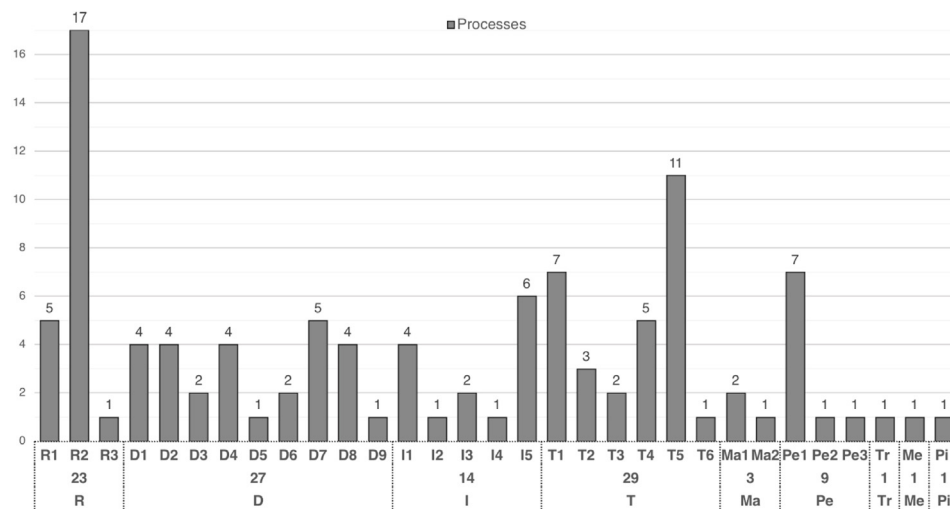


Fig. 5. Number of publications in each of the groups of topics of interest according to the nine processes.

Therefore, in Group **R3**. (tool for gathering accessibility requirements), we identified the following asset: 1 *tool*.

2. For Design (D)

D1. Assets for accessible wearable app design: (1) *design patterns* for accessibility maps development (Liu et al., 2018); (2) *repository* of inclusive mobile application design tools and techniques (Alabi and Gooch, 2011); and (3) *set of guidelines* for accessibility design for wearables (Wentzel et al., 2016a,b).

Therefore, in Group **D1**. (accessible wearable app design), we identified the following assets: 1 *design patterns*, 1 *repository*, and 1 *set of guidelines*.

D2. Assets for adaptive user interfaces: (1) *tool* that aims to improve the creation process as well as the quality of audio-tactile charts (Engel et al., 2019); (2) *MyUI framework* to generate individualized user interfaces and perform adaptations to diverse user needs, devices, and environmental conditions during run time (Peissner et al., 2012); (3) *methods and techniques* for adaptive web accessibility for blind and visually impaired people (state-of-the-art survey) (Raufi et al., 2015); and (4) *techniques* to provide customized accessibility (Oliveira et al., 2018).

Therefore, in Group **D2**. (adaptive user interfaces), we identified the following assets: 1 *tool*, 1 *framework*, 1 *method*, and 2 *techniques*.

D3. Assets for accessibility for specific contexts: (1) *framework* to e-commerce based on an interaction pattern language and on an accessibility guideline targeted to complete vision loss (Junior and Piazza, 2017) and (2) *framework* to help developers make museums more accessible for disabled visitors (Angkananon et al., 2015).

Therefore, in Group **D3**. (accessibility for specific contexts), we identified the following assets: 2 *frameworks*.

D4. Assets for design for disabilities: (1) *prototype* using adaptation techniques to provide the required user experience based on users level of visual impairment (Ferati et al., 2012); (2) *design guidelines* to the Web based application for the physically challenged persons, on the mobile handsets (Johari and Kaur, 2013); (3) *VERITAS framework* to simulate and analyze how users with various impairments interact with many products. It allows needs and preferences of users be accessed by participating applications and devices in order to customize themselves (Scott et al., 2015); and (4) *guidelines* to accessible design for maximal inclusivity. They include an outline of the ways people with disabilities use technology, specific steps to create accessible materials, considerations for non-digital materials and events, and use cases that highlight the benefits of accessible design (Rothberg, 2019).

Therefore, in Group **D4**. (design for disabilities), we identified the following assets: 1 *prototype*, 2 *sets of guidelines*, and 1 *framework*.

D5. Asset for ontology knowledge modeling: (1) *ontology* to model accessibility requirements of learning objects (Ingavéz-Guerra et al., 2018).

Therefore, in Group **D5**. (ontology knowledge modeling), we identified the following assets: 1 *ontology*.

D6. Assets for new representation (design) of web pages: (1) *guidelines* for the design of websites stemming from the NTC 5854 standard aiming at providing accompaniment to software designers regarding the implementation process of accessible web portals (Idrobo et al., 2017) and (2) *framework* to improve the representation of heterogeneous Web pages adapting the document's contents and presentation so that it best fits the blind user's needs (Semaan et al., 2013).

Therefore, in Group **D6**. (new representation – design – of web pages), we identified the following assets: 1 *set of guidelines* and 1 *framework*.

D7. Assets for architecture for accessible web pages: (1) *guidelines* to specify and design the different disabilities mapped into the mobile and accessibility constraints (Chalkia and Bekiaris, 2011); (2) *conceptual software architecture* for the development of accessible web maps for visually impaired users (Calle-Jimenez et al., 2018); (3) *method* to model Web accessibility by moving from abstract to concrete architectural views using aspect-orientation (Martín et al., 2011); (4) *architecture* of CAN (Composable Accessibility Infrastructure), a crowdsourcing infrastructure that collects web accessibility issues and their fixes, dynamically composes solutions on-the-fly, and delivers the crowd-sourced content as teaching materials (Huang et al., 2015); and (5) *method* to integration of accessibility requirements into design primitives of the user interface development method, considering navigation in the Task Model, Abstract User Interface Model and Transformation Model (Miñón et al., 2014).

Therefore, in Group **D7**. (architecture for accessible web pages), we identified the following assets: 1 *set of guidelines*, 2 *architectures*, and 2 *methods*.

D8. Assets for design patterns: (1) *accessibility design patterns* for users with visual disabilities – Authentication adaptor, Blindness adaptor, Dichromatic color vision adaptor, and Blurry vision adaptor (Sanchez-Gordon et al., 2019); (2) *design pattern language* for accessibility (Fogli et al., 2014); (3) *accessibility patterns* for smart environments (Zaki and Forbrig, 2011); and (4) *Non-Functional Requirements Patterns catalog* (Engiel et al., 2014).

Therefore, in Group **D8**. (design patterns), we identified the following assets: 4 *patterns*.

D9. Asset for model-driven engineering: (1) *model driven method* that aims to assist local and worldwide communities of developers throughout the development process of accessible user interfaces (Bouraoui and Gharbi, 2019).

Therefore, in Group **D9**. (model-driven engineering), we identified the following asset: 1 *method*.

3. For Implementation (I)

I1. Assets for techniques to implement accessible products: (1) *method* to facilitate accessibility, *activities* or *techniques* that are used to assist developers (Crabb et al., 2019); (2) *framework* composed of a multidimensional descriptive view of accessibility in ubiquitous learning systems (Salah et al., 2017); (3) *framework* for programming languages that would allow rapid development of such interfaces (Jain and Singh, 2011); and (4) *tool* that automatically increases webpage accessibility by applying three different techniques: link enrichment, image enrichment, and navigation enrichment (Ferati and Sulejmani, 2016).

Therefore, in Group **I1**. (techniques to implement accessible products), we identified the following assets: 1 *method*, 2 *frameworks*, and 1 *tool*.

I2. Asset for adaptive user interfaces implementation: (1) *framework* for increased accessibility through automatically generated adaptive user interfaces (Peissner et al., 2012).

Therefore, in Group **I2**. (adaptive user interfaces implementation), we identified the following asset: 1 *framework*.

I3. Assets for library of software code: (1) *method* to realize accessibility of JavaFX Script applications and improve accessibility of RIAs (Shirogane et al., 2011) and (2) *JavaScript Library* for scalable development of accessible systems (Basman et al., 2011).

Therefore, in Group **I3**. (library of software code), we identified the following assets: 1 *method* and 1 *library*.

I4. Asset for accessibility and standards: (1) *guidelines* for the construction of websites stemming from the NTC 5854 standard aiming at providing accompaniment to software developers regarding the implementation process of accessible web portals (Idrobo et al., 2017);

Therefore, in Group **I4**. (accessibility and standards), we identified the following asset: 1 *set of guidelines*.

I5. Assets for framework to build accessible software: (1) *CAN (Composable Accessibility framework)*, that collects web accessibility issues and their fixes, dynamically composes solutions on-the-fly, and delivers the crowd-sourced content (Huang et al., 2015); (2) *framework* that enforces the accessibility of products and services by enabling automatic simulated accessibility assessment at all the stages of the development (Kaklanis et al., 2011); (3) *framework* for the development of localized web accessibility guidelines (Alayed et al., 2016); (4) *Homero framework* to automate the generation of web pages in accordance with guidelines defined in WCAG 2.0 (Oliveira, 2013); (5) *framework* composed of several standards-based components focused on supporting inclusive learning at Higher Education (Rodríguez et al., 2017); (6) *ACCESS Framework*, an accessibility engine designed to provide cross-platform accessibility support through plug-ins (Heron et al., 2013b).

Therefore, in Group **I5**. (framework to build accessible software), the identified the following assets: 6 *frameworks*.

4. For Testing (T)

T1. Assets for evaluation tools: (1) *Report* about effectiveness of six state-of-the-art tools analyzing their coverage, completeness, and correctness with regard to WCAG 2.0 conformance (Vigo et al., 2013); (2) *Waat tool* for Web content accessibility evaluation regarding different disability profiles (impairments, personas) as well as assistive technologies and devices (Oikonomou et al., 2011a); (3) *web accessibility evaluation report* of cloud-based e-commerce platforms websites (Sohaib et al., 2018); (4) *process* for automatic combination of testing reports for the accessibility of Web applications, obtained by different testing tools and applying different standards on Web accessibility (Hilera et al., 2018); (5) *plug-in* for web content editor to support the web content accessibility manual evaluation process for learning objects in the context of a virtual learning environment (Avila et al., 2012b); (6) *Accessibility Analyzer Tool* to analyze government websites and generate the results automatically (Dongaonkar et al., 2017); and (7) *guidelines* to evaluate a touch surface and a web-based collaboration platform for collaboration (Chiplunkar et al., 2019).

Therefore, in Group **T1**. (evaluation tools), we identified the following assets: 2 *reports*, 2 *tools*, 1 *process*, 1 *plug-in*, and 1 *set of guidelines*.

T2. Assets for automatic detection of accessibility problems: (1) *process* towards the combination of automated and user testing in the form of crowd sourcing to enhance e-Accessibility (Snarud et al., 2014); (2) *Social4all tool* which allows a set of accessibility problems to be solved without modifying the original page code; and (3) *API* that automatically runs tests, analyzing whether the interfaces of an application conform to the specific rules and offer tips for improving accessibility (Moura et al., 2017).

Therefore, in Group **T2**. (automatic detection of accessibility problems), we identified the following assets: 1 *process*, 1 *tool*, and 1 *API*.

T3. Assets for mobile apps testing: (1) *MATE tool* (*Mobile Accessibility Testing*), which automatically explores apps while applying different checks for accessibility issues related to visual impairment (Eler et al., 2018) and (2) *UI Automator Viewer tool* for evaluation of applications accessibility in Android environment (Patil et al., 2016).

Therefore, in Group **T3**. (mobile apps testing), we identified the following assets: 2 *tools*.

T4. Assets for framework for accessibility evaluation/tests: (1) *method* based on web metrics for accessibility evaluation of hospital Websites (Acosta-Vargas et al., 2018); (2) *CHECORSE* – *CHECKER* and *CORRECTOR* tool for ASP *SERVER*-side scripting language reporting and suggesting solutions for the elements in the server source pages (Tuan et al., 2012); (3) *method* for the evaluation of the accessibility and usability of Open Course Ware sites, as well as a *framework* for improving their accessibility and usability (Rodríguez et al., 2017); (4) *evaluation framework* for performing Web accessibility evaluations in different environments (Fernandes et al., 2011); and (5) *framework* that has adopted the latest web standard and supports many assessment configurations to meet the user needs (Oikonomou et al., 2011b).

Therefore, in Group **T4**. (framework for accessibility evaluation/tests), we identified the following assets: 1 *tool*, 2 *methods*, and 3 *frameworks*.

T5. Assets for methods for accessibility evaluation: (1) *method* to determine whether the video content complies with WCAG 2.1 (Acosta-Vargas et al., 2019a); (2) *method* for the web content accessibility evaluation process by applying the Case Based Reasoning technique (Avila et al., 2012b); (3) *method* to evaluate accessibility in mobile applications considering a heuristic method and the WCAG 2.1. (Acosta-Vargas et al., 2019b); (4) *method* to incorporate accessibility requirements and evaluation tools (Sánchez-Gordón et al., 2016); (5) *method* for testing accessibility requirements in RIA (Watanabe et al., 2017, 2012); (6) *WS accessibility assessment Framework (WSaaF)*, developed on the basis of accessibility guidelines, dealing with accessibility issues that can appear both on the presentation level of content and on the content level itself (Giakoumis et al., 2014); (7) *method* based on URL clustering for web accessibility evaluation (Zhang et al., 2015); (8) *method* for heuristic evaluation of web accessibility oriented to types of disabilities, consisting of five stages that allow establishing the level of accessibility and specific problems of a website (Orozco et al., 2016); (9) *method* to creating and evaluating a new version of an application that had no accessibility at all (Ferreira et al., 2019); and (10) *method* which allows to specify the required values for the simulation by recording the interaction with the application, annotating the records and binding the records to the task models (Malý et al., 2012).

Therefore, in Group **T5**. (methods for accessibility evaluation), we identified the following assets: 9 *methods* and 1 *framework*.

T6. Asset for evaluation/tests and standards: (1) *method* to measure the fulfillment of the WCAG 2.0. It also allows estimation about effort to change an existing web content to fulfill the WCAG 2.0 guidelines (Wille et al., 2016).

Therefore, in Group **T6**. (evaluation/tests and standards), we identified the following asset: 1 *method*.

5. For Maintenance (Ma)

Ma1. Assets for refactoring technique: (1) *Client-Side Web Refactoring method* allows for the automatic creation of different and personalized views of the same application to solve the particular bad smells that each user recognizes (Garrido et al., 2012) and (2) *method* and *tool* that can improve web accessibility automatically

(automated refactoring) and improve the accessibility faster than refactoring manually (Ikhshan and Candra, 2018).

Therefore, in Group **Ma1**. (refactoring technique), we identified the following assets: 2 *methods* and 1 *tool*.

Ma2. Asset for reengineering: (1) *method* to promote web accessibility in legacy systems through the use of software awareness. They use self-adaptation strategies to provide customized accessibility (Oliveira et al., 2018).

Therefore, in Group **Ma2**. (reengineering), we identified the following asset: 1 *method*.

6. For Process Establishment (Pe)

Pe1. Assets for accessibility and Software Engineering: (1) *process* to develop an accessible website for Government domain (Conway and Fitzpatrick, 2017); (2) *method* for web sites development which takes into account accessibility expectations of end-users. It relies on an iterative process of three main phases: analysis, design, and evaluation (Reichling and Cherfi, 2013); (3) *process* in which accessibility is present throughout the software lifecycle, based on main phases present in most framework development models (Silva et al., 2017); (4) *method* that integrates accessibility to the Software Engineering process and *Acero tool*, which provides computational support to the proposed approach (Andrade et al., 2018); (5) *disability-aware Software Engineering process model* which considers the needs of people with disabilities, hence improving accessibility an usability of the designed system (Nganji and Nggada, 2011); (6) *process* composed by web design process, organizational factors, personal factors, and external factors for web accessibility development (Velleman et al., 2017); and (7) *process-driven approach* to accessibility involving people with disabilities in the product development process, and establishing partnerships with accessibility experts (Horton and Sloan, 2014).

Therefore, in Group **Pe1**. (accessibility and Software Engineering), we identified the following assets: 5 *processes*, 1 *framework*, 1 *tool*, and 2 *methods*.

Pe2. Asset for case tool for accessibility and Software Engineering: (1) *AccTrace, a CASE tool (Eclipse plug-in)* that uses an ontology to specify the technical implementation of accessibility and promotes traceability of accessibility requirements from conception to the coding phases (Branco, 2013).

Therefore, in Group **Pe2**. (case tool for accessibility and Software Engineering), we identified the following asset: 1 *plug-in*.

Pe3. Asset for accessible software processes: (1) *method* and *tool* considering process description, process workflow, and work products (García-Borgoñón et al., 2014).

Therefore, in Group **Pe3**. (accessible software processes), we identified the following assets: 1 *method* and 1 *tool*.

7. For Training (Tr)

Tr. Asset for gamification for accessibility training: (1) *framework* using gamification as a method for engaging and motivating web designers to increase the adoption of the WCAG 2.0 guidelines (Spyridonis et al., 2017).

Therefore, in Group **Tr**. (gamification for accessibility training), we identified the following asset: 1 *framework*.

8. For Measurement (Me)

Me. Asset for measure the fulfillment of the WCAG 2.0: (1) *method* for a metric to put the results of all 61 success criteria together and calculate a comparable result for the WCAG 2.0 fulfillment (Wille et al., 2016).

Therefore, in Group **Me**. (measure the fulfillment of the WCAG 2.0), we identified the following asset: 1 *method*.

9. For Process Improvement (Pi)

Pi. Asset for capability maturity of usability and accessibility: (1) *Capability maturity model framework*, named MODECUA, to

Table 10
Domains focused on papers.

| Domains | References |
|--------------------------------|--|
| Government | Schulz and Fritsch (2014) Dongaonkar et al. (2017) Conway and Fitzpatrick (2017) Velleman et al. (2017) |
| Education and learning systems | Ludi (2012) Ingavélez-Guerra et al. (2018) Salah et al. (2017) Rodríguez-Ascaso et al. (2011) Avila et al. (2012a) Acosta-Vargas et al. (2019a) |
| E-commerce | Junior and Piazza (2017) Sohaib et al. (2018) |
| Museums | Angkananon et al. (2015) |
| Health | Acosta-Vargas et al. (2018) |
| Air quality | Acosta-Vargas et al. (2019b) |
| Games | Cairns et al. (2019) |
| Smart environment | Zaki and Forbrig (2011) |
| Maps | Liu et al. (2018) Calle-Jimenez et al. (2018) |

evaluate development processes based on usability and accessibility (Quintal and Macías, 2018).

Therefore, in Group **PI**, (capability maturity of usability and accessibility), we identified the following asset: 1 *framework*.

5. Discussion

The SLR presented significant findings that delineate how the proposals of methods, techniques, processes, approaches, and assets to support the development of accessible applications has progressed in the last decade.

In this discussion, we highlight the significance of the main findings identified, comparing the evolution of the field, primary contributions, and research directions to address gaps identified.

5.1. Contexts and platforms

The first aspect we highlight in this discussion is the context in which the studies focused. Few selected papers deal with specific contexts, as shown in Table 10. It is worth noting that most of the studies are devoted to the domains of government and education. However, there have been more studies focusing on other areas, such as games, smart environments, museums, and maps. Research on accessibility has had a strong influence of local, national and international legislation, with a substantial impact on the areas of government and education. However, more research must be applied to other sectors, showing the importance of producing accessible software for a range of systems in a broad scope of systems.

Regarding the platforms, most of the studies focused on Web systems, i.e., 52 papers focused on web, 13 papers focused on mobile devices, and 29 papers presented platform-independent approaches. The main programming languages used to implement tools and frameworks are Java (10 papers), HTML (5 papers), JavaScript (3 papers), Android (3 papers), XML (2 papers), and PHP (2 papers). It is interesting to identify the main models used as background by authors of the selected papers (Fig. 6). We can note the predominance of WCAG, which has been established as the primary reference concerning accessibility guidelines and design issues. It is also extensively used as a reference model in the development of accessibility assessment tools. This study showed a trend in adopting other process-oriented frameworks

to approach accessibility in Software Engineering, as well as other resources, such as design patterns. The proposal of a wider variety of supporting resources for developers and designers is crucial to help enhance accessibility, and to overcome inherent limitations of issues that guidelines may not be able to help.

Despite the substantial uptake in the use of mobile applications, studies concerning proposals for mobile applications are still more limited in comparisons to studies on Web accessibility, for example. This result shows the importance of devoting continued effort to investigate methods, tools, and assets to support the development of accessible mobile applications, considering the technical features of this platform.

However, it is noteworthy that many studies on techniques to support the development of accessible mobile applications has received attention from the research community. In comparison, the WCAG only incorporated specific guidelines focused on mobile devices in its version 2.1 in 2018 (World Wide Web Consortium, 2019c).

5.2. Evolution of research on accessibility and Software Engineering

Considering the three SLR and concerning publications per year, we can observe that there was an average of 9.59 articles published per year. Regarding the number of articles published per year, there was a similarity in the three SLR. However, the main exceptions were the following three periods: 1998–2003, 2013, and 2015, as presented in Fig. 7.

Freire et al. (2007a) noticed from 2004 to 2006 that the number of studies on techniques for accessibility had increased considerably. This result was reinforced by Dias et al. (2010), especially considering the years of 2007 and 2009. Considering the 2010–2019 period, we can observe more stability concerning the number of publications.

We can also detect the evolution of publications on accessibility in each Software Engineering phase. As presented in Fig. 8, few processes, such as software requirements and implementation, had improved their number of publications. The testing process remained more stable and concentrated the most significant amount of publications over time. The rise of 9 articles in the process establishment area was a remarkable result, which indicates a change and concern with accessibility.

In comparison with the number of papers analyzed in the previous SLR, we noticed that the past decade has still devoted considerable effort to devise approaches for accessible software development. However, there have been fewer recent studies per year in comparison to periods such as the years of 2007 and 2009.

5.3. Software development phases and activities approached by the studies

The first SLR, carried out by Freire et al. (2007a), indicated that several development activities were poorly addressed by scientific research efforts. The authors concluded that software testing was the primary process focused by researchers. The second SLR, carried out by Dias et al. (2010), concluded that research on web accessibility has grown since 2007 and, similarly, software testing was the main process focused by researchers.

The third SLR, presented in this paper, indicated interesting and original results. As presented in Fig. 2, the testing process remains the main focus of the researchers. However, design and software requirements phases had similar results and can also be highlighted. The process establishment and the process improvement were identified in this SLR only, and we can conclude that researchers began to associate accessibility with software processes as a way to contribute to the quality of the final product.

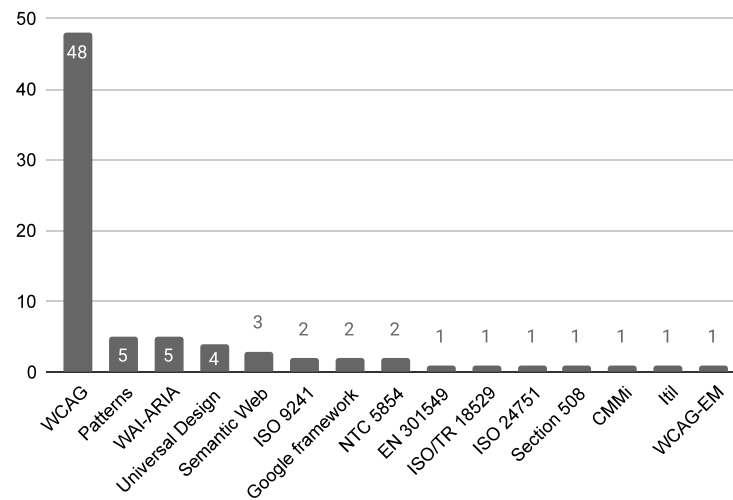


Fig. 6. Main models used as background.

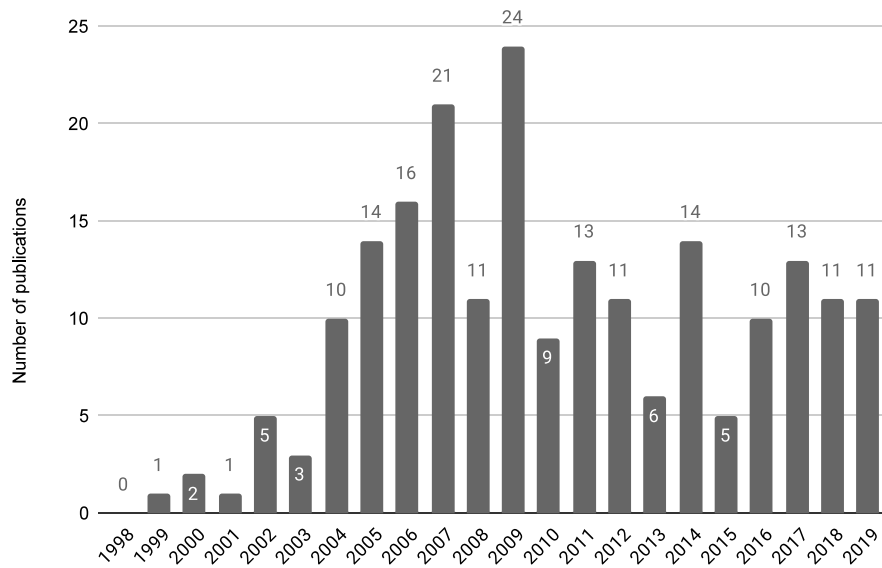


Fig. 7. Number of publications per year considering results of the three SLR.

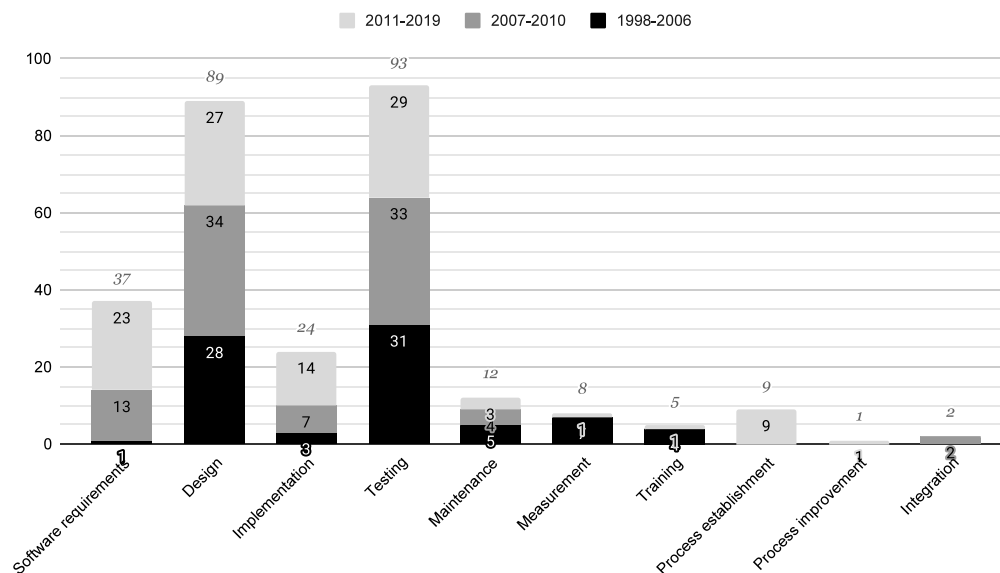


Fig. 8. Number of publications considering results of three SLR considering software development phases and other activities.

Additionally, Freire et al. (2007a) identified in their study that investigation on interface design patterns would be a promising research field in HCI. Dias et al. (2010) did not identify studies regarding this subject. In our findings, four studies were obtained, presenting design patterns to RIA (Fogli et al., 2014), smart environments (Zaki and Forbrig, 2011), software requirements elicitation (Engiel et al., 2014), and software process (Sanchez-Gordon et al., 2019).

In comparison with the two previous SLR studies, the present study identified a significant growth in studies approaching proposals for requirements elicitation (R2 in Fig. 5). This finding may indicate that research on development techniques for accessibility has incorporated many principles of User-Centered Design and the involvement of users with disabilities. This observation is very relevant when comparing the findings of the first SLR (Freire et al., 2007a), for example. In that study, a large number of studies published between 1998 and 2006 were devoted to developing automated tools to test technical aspects of conformance to accessibility guidelines. Despite the importance of such tools in the development process, there was a severe lack of techniques that addressed human aspects of the interaction of people with disabilities with interactive systems. It is very positive that the number of such studies has had growth.

We could observe a growing emphasis on research on patterns and ontologies over previous reviews, and for the first time, a refactoring technique for accessibility was proposed. It is also interesting to note the inclusion of the accessibility subject in specific journals and conferences, such as fault tolerance (Eler et al., 2018) and gamification (Spyridonis et al., 2017). Indeed, game developers are interested in implementing accessibility features (Porter and Kientz, 2013) and specific guidelines are now provided to indicate how games may be accessible (AbleGamers, 2019). Additionally, accessibility related to mobile was also reinforced. In particular, new theories and tools related to testing and evaluating mobile apps were published in the last years.

The proposal of approaches for measurement of accessibility has decreased significantly in comparison to the previous SLR studies in the past decades. Until the first years of the 2010 decade, there were many studies devoted to the proposal and analysis of accessibility measures (Vigo and Brajnik, 2011). These studies proposed different formulas to calculate indicators to quantify conformance to web accessibility guidelines. Many of those studies employed solely automated evaluation tools, which may compromise the validity of the results. There was little investigation into the correlation between these indicators and results from studies involving users with disabilities. Future studies on accessibility measures would be needed to analyze the validity of existing measures and improvements to allow the effective use of those approaches in real projects, such as in public policy and law enforcement, for example.

Another positive aspect observed was the growth in studies covering process improvement aiming at better accessibility. Achieving good accessibility involves developing software that helps users with disabilities to perform their tasks and achieve their goals. In this sense, developing systems from a User-Centered Design approach is essential, and it is crucial to have a process-oriented view of this development, including the involvement of users in different stages. The growth in research to enable development teams to incorporate accessibility in their process and the process improvement is in line with other tendencies in companies and governmental bodies. The United Kingdom government, for example, has proposed guides to implement processes to improve the accessibility of governmental digital services.⁶ The European Union has also developed

recommendations, such as Accessibility Procurement toolkits, contract management processes and other, such as the initiatives under the Mandate 376.⁷ Despite the identification of some initial work in this area, investigating the proposal or process improvement tools and techniques to address accessibility is a significant challenge to be tackled, in order to provide research evidence to improve practice in contexts such as government and commercial solutions.

6. Conclusion

This paper presented a SLR about accessibility in the context of software development processes. It was an update of two SLRs previously carried out.

This study showed that software testing and software design had been the focus of research in the last decade when considering the inclusion of accessibility during software development. There was also a slight increase in the number of publications from 2011 to 2019 compared to previous SLRs.

Based on the results presented in this paper, it is possible to highlight open questions for future studies. Nowadays, the focus is on complete or partial visual impairment, and few papers discussing other disabilities, such as hearing and cognitive disabilities, were found. For example, this SLR has identified requirements engineering techniques, ontology, frameworks, and testing tools directed at the visual impairment, and we could note the gap regarding other deficiencies.

Considering the software design phase, we did not find a reference architecture in the software accessibility domain. It would be useful to help accelerate and ensure affordable software. A reference architecture could contribute to solving two problems identified by Crabb et al. (2019): “*lack of understanding in how a person with a disability uses technology impacts on how technology interactions are designed*” and “*developers require easy methods of implementing accessibility features to increase overall technology accessibility*”.

Few articles are related to software maintenance. Considering that software currently available in general is not accessible and that the government of many countries requires accessibility through laws, there is a lack of methods and tools to correct the problems.

We noticed that there are initiatives and many challenges on some topics. About games, for example, studies are focusing on requirements and vocabulary, and actual demand is to evolve to design, development, and test of accessible games considering different disabilities. Also, games for elderly users should be a research area for the next years.

Finally, we can also mention the incorporation of accessibility to agile methodologies and open source development. Currently, startups (and small companies) develop several mobile apps, websites, and others adopting extensively agile methods. Similarly, open-source software has an essential role for final users and corporations. Including accessibility in both cases is crucial to ensure the accessibility of the generated products.

CRedit authorship contribution statement

Débora Maria Barroso Paiva: Investigation, Methodology, Writing - original draft, Supervision, Project administration. **André Pimenta Freire:** Investigation, Conceptualization, Methodology, Writing - review & editing. **Renata Pontin de Mattos Fortes:** Methodology, Writing - original draft, Visualization, Validation, Project administration.

⁶ Available online at <https://www.gov.uk/service-manual/helping-people-to-use-your-service/making-your-service-accessible-an-introduction>, last accessed on 20th July 2020.

⁷ Available online at <http://mandate376.standards.eu/>, last accessed on 20th July 2020.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the São Paulo Research Foundation – FAPESP, Brazil [grant #2018/08195-9 (SPRINT)] and by the CNPq-MCTI; FAPESP, Brazil [grant #2015/24525-0].

References

- AbleGamers, 2019. Accessible player experiences (APX). URL <https://accessible.games/accessible-player-experiences/>.
- Acosta-Vargas, P., Acosta, T., Luján-Mora, S., 2018. Framework for accessibility evaluation of Hospital Websites. In: 2018 International Conf. on EDemocracy & EGovernment (ICEDEG). IEEE, pp. 9–15.
- Acosta-Vargas, P., Salvador-Ullauri, L., Pérez-Medina, J.L., Rybarczyk, Y., 2019a. Accessibility evaluation of multimedia resources in selected latin america universities. In: 2019 Sixth International Conf. on EDemocracy & EGovernment (ICEDEG). IEEE, pp. 249–255.
- Acosta-Vargas, P., Salvador-Ullauri, L., Pérez-Medina, J.L., Zalakeviciute, R., Hernandez, W., 2019b. Heuristic method of evaluating accessibility of mobile in selected applications for air quality monitoring. In: International Conf. on Applied Human Factors and Ergonomics. Springer, pp. 485–495.
- Alabi, H.I., Gooch, B., 2011. The accessibility toolkit. In: Proceedings of the 10th SIGPLAN Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software. pp. 145–148.
- Alayed, A., Wald, M., Draffan, E., 2016. A framework for the development of localised web accessibility guidelines for University Websites in Saudi Arabia. In: International Conf. on Universal Access in Human-Computer Interaction. Springer, pp. 3–13.
- Andrade, W.T., Branco, R.G., Cagnin, M.I., Paiva, D.M.B., 2018. Incorporating accessibility elements to the software engineering process. *Advances in Human-Computer Interaction* 1–17.
- Angkananon, K., Wald, M., Gilbert, L., 2015. Technology enhanced interaction framework and method for accessibility in Thai Museums. In: 2015 3rd International Conf. on Information and Communication Technology (ICoICT). IEEE, pp. 316–321.
- Avila, C., Baldiris, S., Fabregat, R., Guevara, J.C., 2012a. A web content accessibility evaluation process for learning objects in the context of a virtual learning environment. In: International Conf. on Web-Based Learning. Springer, pp. 181–190.
- Avila, C., Baldiris, S., Fabregat, R., Guevara, J.C., 2012b. Accessibility evaluation improvement using case based reasoning. In: 2012 Frontiers in Education Conf. Proceedings. IEEE, pp. 1–6.
- Basman, A.M., Lewis, C.H., Clark, C.B., 2011. To inclusive design through contextually extended IoC: Infusion IoC, a JavaScript library and mentality for scalable development of accessible and maintainable systems. In: Proceedings of the ACM International Conf. Companion on Object Oriented Programming Systems Languages and Applications Companion. pp. 237–256.
- Bouraoui, A., Gharbi, I., 2019. Model driven engineering of accessible and multiplatform graphical user interfaces by parameterized model transformations. *Sci. Comput. Program.* 172, 63–101.
- Branco, R.G., 2013. Acessibilidade nas Fases de Engenharia de Requisitos, Projeto e Codificação de Software: Uma Ferramenta de Apoio (Master's thesis). Universidade Federal de Mato Grosso do Sul, Brazil.
- Branco, R.G., Cagnin, M.I., Paiva, D.M.B., 2014. AccTrace: Accessibility in phases of requirements engineering, design, and coding software. In: 2014 14th International Conf. on Computational Science and Its Applications. IEEE, pp. 225–228.
- Cairns, P., Power, C., Barlet, M., Haynes, G., 2019. Future design of accessibility in games: A design Vocabulary. *Int. J. Hum.-Comput. Stud.* <http://dx.doi.org/10.1016/j.ijhcs.2019.06.010>.
- Calle-Jimenez, T., Eguéz-Sarzosa, A., Luján-Mora, S., 2018. Design of an architecture for accessible web maps for visually impaired users. In: International Conf. on Applied Human Factors and Ergonomics. Springer, pp. 221–232.
- Chalkia, E., Bekiaris, E., 2011. A harmonised methodology for the components of Software Applications accessibility and its evaluation. In: International Conf. on Universal Access in Human-Computer Interaction. Springer, pp. 197–205.
- Chiplunkar, S., Maini, A., Ram, D., Zheng, Z., Zheng, Y., 2019. Drawxi: an accessible drawing tool for collaboration. In: CHI Conf. on Human Factors in Computing Systems. pp. 1–6.
- Conway, V., Fitzpatrick, K., 2017. Creating accessible local government: The process. In: Proceedings of the 14th Web for All Conf. on the Future of Accessible Work. pp. 1–4.
- Crabb, M., Heron, M., Jones, R., Armstrong, M., Reid, H., Wilson, A., 2019. Developing accessible services: Understanding current knowledge and areas for future support. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. In: CHI'19, ACM, New York, NY, USA, pp. 1–12. <http://dx.doi.org/10.1145/3290605.3300446>.
- Crespo, R.G., Espada, J.P., Burgos, D., 2016. Social4all: Definition of specific adaptations in web applications to improve accessibility. *Comput. Stand. Interfaces* 48 (C), 1–9. <http://dx.doi.org/10.1016/j.csi.2016.04.001>.
- Dias, A.L., Fortes, R.P.M., Masiero, P.C., 2012. Increasing the quality of web systems: by Inserting requirements of Accessibility and Usability. In: 2012 Eighth International Conf. on the Quality of Information and Communications Technology. IEEE, Lisbon, Portugal, pp. 224–229.
- Dias, A.L., Fortes, R.P.M., Masiero, P.C., Goularte, R., 2010. A systematic review on the insertion of accessibility in the development phases of web software engineering. In: Proceedings of the Symposium of Human Factors in Computational Systems - IHC. Belo Horizonte - MG - Brazil, pp. 39–48.
- Dongaonkar, S.U., Vadali, R.S., Dhutadmal, C., 2017. Accessibility analyzer: Tool for new adaptations in government web applications to improve accessibility. In: 2017 International Conf. on Computing, Communication, Control and Automation (ICCUBEA). IEEE, pp. 1–5.
- Eler, M.M., Rojas, J.M., Ge, Y., Fraser, G., 2018. Automated accessibility testing of mobile Apps. In: 2018 IEEE 11th International Conf. on Software Testing, Verification and Validation (ICST). IEEE, pp. 116–126.
- Engel, C., Müller, E.F., Weber, G., 2019. SVGPlot: an accessible tool to generate highly adaptable, accessible audio-tactile charts for and from blind and visually impaired people. In: Proceedings of the 12th ACM International Conf. on Pervasive Technologies Related To Assistive Environments. ACM, pp. 186–195.
- Engiel, P., Cappelli, C., do Prado Leite, J.C.S., 2014. Eliciting concepts from the Brazilian access law using a combined approach. In: Proceedings of the 29th Annual ACM Symposium on Applied Computing. pp. 1001–1006.
- Ferati, M., Kurti, A., Vogel, B., Raufi, B., 2016. Augmenting requirements gathering for people with special needs using IoT: a position paper. In: Proceedings of the 9th International Workshop on Cooperative and Human Aspects of Software Engineering. pp. 48–51.
- Ferati, M., Sulejmani, L., 2016. Automatic adaptation techniques to increase the web accessibility for blind users. In: International Conf. on Human-Computer Interaction. Springer, pp. 30–36.
- Ferati, M., Vogel, B., Kurti, A., Raufi, B., Astals, D.S., 2012. Web accessibility for visually impaired people: Requirements and design issues. In: Usability-and Accessibility-Focused Requirements Engineering. Springer, pp. 79–96.
- Fernandes, N., Lopes, R., Carriço, L., 2011. An architecture for multiple web accessibility evaluation environments. In: Stephanidis, C. (Ed.), *Universal Access in Human-Computer Interaction. Design for All and Inclusion*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 206–214.
- Ferreira, M.A.M., Bonacin, R., 2014. Eliciting accessibility requirements for people with hearing loss: a Semantic and Norm analysis. In: International Conf. on Human-Computer Interaction. Springer, pp. 277–288.
- Ferreira, A., do Nascimento, L., Kuroda, V., Zandona, D., dos Anjos, M., Ribeiro, B., Pellegrini, F., Correia, W., Macêdo, J., Quintino, J., 2019. CasaE Xperience-Creation Method focused in accessibility. In: International Conf. on Applied Human Factors and Ergonomics. Springer, pp. 281–290.
- Fogli, D., Provenza, L.P., Bernareggi, C., 2014. A universal design resource for rich internet applications based on design patterns. *Univ. Access Inf. Soc.* 13 (2), 205–226.
- Freire, A.P., Goularte, R., Fortes, R.P.M., 2007a. Survey on Techniques for Development of Accessible Web Applications. Tech. rep., Technical Report n. 303 - In Portuguese, University of São Paulo.
- Freire, A.P., Goularte, R., Fortes, R.P.M., 2007b. Techniques for developing more accessible web applications: a survey towards a process classification. In: Proceedings of the 25th Annual International Conf. on Design of Communication, SIGDOC 2007, El Paso, Texas, USA, October 22–24, 2007. pp. 162–169. <http://dx.doi.org/10.1145/1297144.1297177>.
- Fuggetta, A., 2000. Software process: a Roadmap. In: Proceedings of the Conf. on the Future of Software Engineering. ACM, pp. 25–34.
- Garía-Borgoñón, L., Barcelona, M., Garía-García, J., Escalona, M.J., 2014. Software process accessibility in practice: a case study. *Procedia Comput. Sci.* 27, 292–301.
- Garrido, A., Firmenich, S., Rossi, G., Grigera, J., Medina-Medina, N., Harari, I., 2012. Personalized web accessibility using client-side refactoring. *IEEE Internet Comput.* 17 (4), 58–66.
- Giakoumis, D., Votis, K., Tzouvaras, D., 2014. Introducing web service accessibility assessment techniques through a unified quality of service context. *Serv. Orient. Comput. Appl.* 8 (2), 159–174.
- Gulliksen, J., Harker, S., Vanderheiden, G., 2004. Guidelines, standards, methods and processes for software accessibility. *Univ. Access Inf. Soc.* 3 (1), 1–5. <http://dx.doi.org/10.1007/s10209-003-0068-4>.
- Heron, M., Hanson, V.L., Ricketts, I.W., 2013a. ACCESS: a technical framework for adaptive accessibility support. In: Proceedings of the 5th ACM SIGCHI Symposium on Engineering Interactive Computing Systems. pp. 33–42.

- Heron, M., Hanson, V.L., Ricketts, I., 2013b. Open source and accessibility: Advantages and limitations. *J. Interact. Sci.* 1 (1), 1–10.
- Hilera, J.R., Otón, S., Timbi-Sisalima, C., Aguado-Delgado, J., Estrada-Martínez, F.J., Amado-Salvatierra, H.R., 2018. Combining multiple web accessibility evaluation reports using semantic web technologies. In: Paspallis, N., Raspopoulos, M., Barry, C., Lang, M., Linger, H., Schneider, C. (Eds.), *Advances in Information Systems Development*. Springer International Publishing, Cham, pp. 65–78.
- Horton, S., Sloan, D., 2014. *Accessibility in Practice: a Process-driven approach to accessibility*. In: *Inclusive Designing*. Springer, pp. 105–115.
- Huang, Y., Dobreski, B., Deo, B.B., Xin, J., Barbosa, N.M., Wang, Y., Bigham, J.P., 2015. CAN: Composable accessibility infrastructure via data-driven crowdsourcing. In: *Proceedings of the 12th Web for All Conf.* pp. 1–10.
- Humphrey, W., 1989. *Managing the Software Process*. Addison-Wesley Longman Publishing Co., Inc..
- Idrobo, C.S., Idrobo, M.I., Chan-chí, G.E., Vidal, M.I., 2017. Accessibility guide for the design and implementation of Websites according to the NTC 5854 Standard. In: *Colombian Conf. on Computing*. Springer, pp. 572–585.
- Ikhsan, I.N., Candra, M.Z.C., 2018. Automatically: An automated refactoring method and tool for improving web accessibility. In: *2018 5th International Conf. on Data and Software Engineering (ICoDSE)*. IEEE, pp. 1–6.
- Ingavélez-Guerra, P., Ulloa-Amaya, M., Vera-Rea, A., Galán-Mena, J., Robles-Bykbaev, V., Hilera, J., Otón, S., 2018. An ontological network to identify accessibility metadata in learning objects: an approach based on Web content Accessibility Guidelines, Schemas, and disabilities analysis. In: *Proceedings of the 2018 IEEE International Autumn Meeting on Power, Electronics and Computing*. IEEE, pp. 1–6.
- ISO/IEC, 1998. *ISO/IEC 12207 Standard for Informational Technology, Software Lifecycle Processes*. ISO/IEC, 1, ch. de la Voie-Creuse - CP 56 - CH-1211 Geneva 20 - Switzerland.
- ISO/IEC/IEEE, 2017. *ISO/IEC/IEEE 12207:2017 systems and software engineering, software life cycle processes*. 1, ch. de la Voie-Creuse - CP 56 - CH-1211 Geneva 20 - Switzerland.
- Ivarsson, M., Gorschek, T., 2011. A method for evaluating rigor and industrial relevance of technology evaluations. *Empir. Softw. Eng.* 16 (3), 365–395.
- Jain, A., Singh, S., 2011. Modified programming language framework for IVRS accessibility of graphical user interfaces. In: *2011 Fourth International Conf. on Ubi-Media Computing*. IEEE, pp. 163–167.
- Johari, K., Kaur, A., 2013. *Web accessibility: Designing and Testing of Web based application for Persons with disabilities*. In: *Computer Networks & Communications (NetCom)*. Springer, pp. 729–736.
- Junior, P.T.A., Piazza, M., 2017. Framework interface components for accessibility issues in E-Commerce. In: *Advances in Ergonomics Modeling, Usability & Special Populations*. Springer, pp. 527–539.
- Kaklanis, N., Moschonas, P., Moustakas, K., Tzovaras, D., 2011. A framework for automatic simulated accessibility assessment in virtual environments. In: *International Conf. on Digital Human Modeling*. Springer, pp. 302–311.
- Kitchenham, B., Charters, S., 2007. *Guidelines for Performing Systematic Literature Reviews in Software Engineering*. Tech. rep., EBSE Technical Report - EBSE-2007-01.
- Kitchenham, B., Pretorius, R., Budgen, D., Pearl Brereton, O., Turner, M., Niazi, M., Linkman, S., 2010. Systematic literature reviews in software engineering - A tertiary study. *Inf. Softw. Technol.* 52 (8), 792–805. <http://dx.doi.org/10.1016/j.infsof.2010.03.006>.
- Krainz, E., Miesenberger, K., Feiner, J., 2018. Can we improve App Accessibility with advanced Development Methods?. In: *Miesenberger, K., Kouroupetroglou, G. (Eds.), Computers Helping People with Special Needs*. Springer International Publishing, Cham, pp. 64–70.
- Liu, Z., Shabani, S., Balet, N.G., Sokhn, M., Cretton, F., 2018. How to motivate participation and Improve Quality of crowdsourcing when building accessibility maps. In: *2018 15th IEEE Annual Consumer Communications & Networking Conf. (CCNC)*. IEEE, pp. 1–6.
- Ludi, S., 2012. Requirements Gathering and domain understanding for assistive technology to support low vision and Sighted Students. In: *Usability-and Accessibility-Focused Requirements Engineering*. Springer, pp. 117–132.
- Ludi, S., Canter, A., Ellis, L., Shrestha, A., 2012. Requirements Gathering for assistive Technology that includes low vision and sighted users. In: *2012 First International Workshop on Usability and Accessibility Focused Requirements Engineering (UsARE)*. IEEE, pp. 25–31.
- Malý, I., Bittner, J., Slavík, P., 2012. Using annotated task models for accessibility evaluation. In: *International Conf. on Computers for Handicapped Persons*. Springer, pp. 315–322.
- Martín, A., Cechich, A., Rossi, G., 2011. Accessibility at early stages: Insights from the designer perspective. In: *Proceedings of the International Cross-Disciplinary Conf. on Web Accessibility*. pp. 1–9.
- Miñón, R., Moreno, L., Martínez, P., Abascal, J., 2014. An approach to the integration of accessibility requirements into a user Interface Development Method. *Sci. Comput. Program.* 86, 58–73. <http://dx.doi.org/10.1016/j.scico.2013.04.005>, Special issue on Software Support for User Interface Description Languages (UIDL 2011).
- Moura, C.J.M., de Oliveira, S., Faria, K.A.C., de Andrade Freitas, E.N., 2017. A new API for android accessibility testing. In: *2017 International Conf. on Computational Science and Computational Intelligence (CSCI)*. IEEE, pp. 594–598.
- Nganji, J.T., Nggada, S.H., 2011. Disability-aware software engineering for improved system accessibility and usability. *Int. J. Softw. Eng. Appl.* 5 (3), 47–62.
- Oikonomou, T., Kaklanis, N., Votis, K., Kastori, G.-E., Partarakis, N., Tzovaras, D., 2011a. WaaT: Personalised web Accessibility Evaluation Tool. In: *Proceedings of the International Cross-Disciplinary Conf. on Web Accessibility*. In: *W4A'11, ACM, New York, NY, USA*, pp. 1–2. <http://dx.doi.org/10.1145/1969289.1969315>.
- Oikonomou, T., Kaklanis, N., Votis, K., Tzovaras, D., 2011b. An accessibility assessment framework for improving designers experience in web applications. In: *International Conf. on Universal Access in Human-Computer Interaction*. Springer, pp. 258–266.
- Oliveira, R.C., 2013. *Homero: Um Framework de Apoio ao Desenvolvimento de Interfaces de Aplicações Web Acessíveis* (Master's thesis). Universidade Federal de Mato Grosso do Sul, Brazil.
- Oliveira, R.C., Freire, A.P., Paiva, D.M.B., Cagnin, M.I., Rubinsztejn, H., 2014. A framework to facilitate the implementation of technical aspects of web accessibility. In: *Stephanidis, C., Antona, M. (Eds.), Universal Access in Human-Computer Interaction*. In: *UAHCI*, Springer International Publishing, Cham, pp. 3–13.
- Oliveira, R.F., da Mota Moura, A.M., Leite, J.C.S.P., 2018. Reengineering for accessibility: a strategy based on software awareness. In: *Proceedings of the 17th Brazilian Symposium on Software Quality*. pp. 180–189.
- Oliveira, R., Silva, L., Leite, J.C.S.P., Moreira, A., 2016. Eliciting accessibility requirements an approach based on the NFR framework. In: *Proceedings of the 31st Annual ACM Symposium on Applied Computing*. pp. 1276–1281.
- Orozco, A., Tabares, V., Duque, N., 2016. Methodology for heuristic evaluation of web accessibility oriented to types of disabilities. In: *International Conf. on Universal Access in Human-Computer Interaction*. Springer, pp. 91–97.
- Patil, N., Bhole, D., Shete, P., 2016. Enhanced UI automator viewer with improved android accessibility evaluation features. In: *2016 International Conf. on Automatic Control and Dynamic Optimization Techniques (ICADOT)*. IEEE, pp. 977–983.
- Peissner, M., Häbe, D., Janssen, D., Sellner, T., 2012. MyUI: Generating accessible user interfaces from multimodal design patterns. In: *Proceedings of the 4th ACM SIGCHI Symposium on Engineering Interactive Computing Systems*. ACM, pp. 81–90.
- Peissner, M., Schuller, A., Ziegler, D., Knecht, C., Zimmermann, G., 2014. Requirements for the successful market Adoption of Adaptive user Interfaces for Accessibility. In: *International Conf. on Universal Access in Human-Computer Interaction*. Springer, pp. 431–442.
- Pellegrini, F., Anjos, M., Florentin, F., Ribeiro, B., Correia, W., Quintino, J., 2019. How to prioritize accessibility in Agile Projects. In: *International Conf. on Applied Human Factors and Ergonomics*. Springer, pp. 271–280.
- Petersen, K., Vakkalanka, S., Kuzniarz, L., 2015. Guidelines for conducting systematic mapping studies in software engineering: An update. *Inf. Softw. Technol.* 64, 1–18. <http://dx.doi.org/10.1016/j.infsof.2015.03.007>.
- Porter, J.R., Kientz, J.A., 2013. An empirical study of issues and barriers to mainstream video game accessibility. In: *Proceedings of the 15th International ACM SIGACCESS Conf. on Computers and Accessibility*. ACM, p. 3.
- Quintal, C., Macías, J.A., 2018. A capability maturity proposal for usability and accessibility centered processes. In: *Proceedings of the XIX International Conf. on Human Computer Interaction*. ACM, pp. 1–8.
- Raufi, B., Ferati, M., Zenuni, X., Ajdari, J., Ismaili, F., 2015. Methods and techniques of adaptive Web accessibility for the Blind and visually impaired. *Procedia-Soc. Behav. Sci.* 195, 1999–2007.
- Reichling, M., Cherfi, S.S.S., 2013. Integrating accessibility as a quality property in Web Developments. In: *IEEE 7th International Conf. on Research Challenges in Information Science (RCIS)*. IEEE, pp. 1–12.
- Rodríguez, G., Pérez, J., Cueva, S., Torres, R., 2017. A framework for improving web accessibility and usability of open course ware sites. *Comput. Educ.* 109, 197–215.
- Rodríguez-Ascaso, A., Botcario, J.G., Finat, C., del Campo, E., Saneiro, M., Alcocer, E., y Restrepo, E.G., Mazzone, E., 2011. Inclusive scenarios to evaluate an open and standards-based framework that supports accessibility and personalisation at higher education. In: *International Conf. on Universal Access in Human-Computer Interaction*. Springer, pp. 612–621.
- Ross, A.S., Zhang, X., Fogarty, J., Wobbrock, J.O., 2017. Epidemiology as a framework for large-scale mobile application accessibility assessment. In: *Proceedings of the 19th International ACM SIGACCESS Conf. on Computers and Accessibility*. pp. 2–11.
- Rothberg, M.A., 2019. Designing for inclusion: Ensuring accessibility for people with disabilities. In: *Consumer Informatics and Digital Health*. Springer, pp. 125–143.

- Salah, N.B., Saâdi, I.B., Ghezala, H.B., 2017. A comprehensive view of u-accessibility in context-aware learning systems for disabled learners. In: 2017 IEEE/ACS 14th International Conf. on Computer Systems and Applications (AICCSA). IEEE, pp. 41–48.
- Sánchez-Gordón, M.L., Moreno, L., 2014. Toward an integration of web accessibility into testing processes. *Procedia Comput. Sci.* 27, 281–291.
- Sánchez-Gordón, S., Sánchez-Gordón, M.L., Luján-Mora, S., 2016. Towards an engineering process for developing accessible software in small software enterprises. In: Proceedings of the 11th International Conf. on Evaluation of Novel Software Approaches to Software Engineering, Portugal, pp. 241–246.
- Sanchez-Gordon, S., Sánchez-Gordón, M., Yilmaz, M., O'Connor, R.V., 2019. Integration of accessibility design patterns with the software implementation process of ISO/IEC 29110. *J. Softw.: Evol. Process* 31 (1), e1987. <http://dx.doi.org/10.1002/smr.1987>, e1987 JSME-18-0100.R1. [arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/smr.1987](https://onlinelibrary.wiley.com/doi/pdf/10.1002/smr.1987).
- Schulz, T., Fritsch, L., 2014. Accessibility and Inclusion requirements for Future e-Identity solutions. In: International Conf. on Computers for Handicapped Persons. Springer, pp. 316–323.
- Scott, M.J., Spyridonis, F., Ghinea, G., 2015. Designing for designers: Towards the development of accessible ICT products and services using the VERITAS Framework. *Comput. Stand. Interfaces* 42, 113–124.
- Semaan, B., Tekli, J., Issa, Y.B., Tekli, G., Chbeir, R., 2013. Toward enhancing web accessibility for blind users through the Semantic Web. In: 2013 International Conf. on Signal-Image Technology & Internet-Based Systems. IEEE, pp. 247–256.
- Shirogane, J., 2014. Support Method to Elicit accessibility requirements. In: Requirements Engineering. Springer, pp. 210–223.
- Shirogane, J., Kato, T., Hashimoto, Y., Tachibana, K., Iwata, H., Fukazawa, Y., 2011. Method to improve accessibility of rich internet applications. In: Symposium of the Austrian HCI and Usability Engineering Group. Springer, pp. 349–365.
- Silva, J.S., Gonçalves, R., Pereira, A., 2017. Accessibility in the software life cycle a maieutic exercise in software engineering. In: 2017 12th Iberian Conf. on Information Systems and Technologies (CISTI). IEEE, pp. 1–5.
- Sirikitsathian, P., Chaveesuk, S., Sathitwiriawong, C., 2017. A conceptual framework for better understanding of factors influencing accessibility to a Website and its acceptance by university students with visual impairments. In: 2017 9th International Conf. on Information Technology and Electrical Engineering (ICITEE). IEEE, pp. 1–5.
- Snaprud, M., Rasta, K., Andreasson, K., Nietzio, A., 2014. Benefits and challenges of combining automated and user testing to enhance e-accessibility—the european internet inclusion initiative. In: International Conf. on Computers for Handicapped Persons. Springer, pp. 137–140.
- Sohaib, O., Naderpour, M., Hussain, W., 2018. SaaS E-Commerce platforms web accessibility evaluation. In: 2018 IEEE International Conf. on Fuzzy Systems (FUZZ-IEEE). IEEE, pp. 1–7.
- Spyridonis, F., Daylamani-Zad, D., Paraskevopoulos, I.T., 2017. The Gamification of accessibility design: A proposed framework. In: 2017 9th International Conf. on Virtual Worlds and Games for Serious Applications (Vs-Games). IEEE, pp. 233–236.
- Tuan, D.T., Phan, V.-H., et al., 2012. Checking and correcting the source code of web pages for accessibility. In: 2012 IEEE RIVF International Conf. on Computing & Communication Technologies, Research, Innovation, and Vision for the Future. IEEE, pp. 1–4.
- Velleman, E.M., Nahuis, I., van der Geest, T., 2017. Factors explaining adoption and implementation processes for web accessibility standards within e-government systems and organizations. *Univ. Access Inf. Soc.* 16 (1), 173–190.
- Vigo, M., Brajnik, G., 2011. Automatic web accessibility metrics: Where we are and where we can go. *Interact. Comput.* 23 (2), 137–155.
- Vigo, M., Brown, J., Conway, V., 2013. Benchmarking web accessibility evaluation tools: Measuring the harm of sole reliance on automated tests. In: Proceedings of the 10th International Cross-Disciplinary Conf. on Web Accessibility. In: W4A'13, ACM, New York, NY, USA, pp. 1–10. <http://dx.doi.org/10.1145/2461121.2461124>, URL <http://doi.acm.org/10.1145/2461121.2461124>.
- Watanabe, W.M., Fortes, R.P.M., Dias, A.L., 2012. Using acceptance tests to validate accessibility requirements in RIA. In: Proc. of the International Cross-Disciplinary Conf. on Web Accessibility. In: W4A 2012, ACM, New York, NY, USA, pp. 1–10. <http://dx.doi.org/10.1145/2207016.2207022>.
- Watanabe, W.M., Fortes, R.P., Dias, A.L., 2017. Acceptance tests for validating ARIA requirements in Widgets. *Univ. Access Inf. Soc.* 16 (1), 3–27.
- Wentzel, J., Velleman, E., van der Geest, T., 2016a. Developing accessibility design guidelines for wearables: Accessibility standards for multimodal wearable devices. In: International Conf. on Universal Access in Human-Computer Interaction. Springer, pp. 109–119.
- Wentzel, J., Velleman, E., van der Geest, T., 2016b. Wearables for all: Development of guidelines to stimulate accessible wearable technology design. In: Proceedings of the 13th Web for All Conf. In: W4A'16, Association for Computing Machinery, New York, NY, USA, pp. 1–4. <http://dx.doi.org/10.1145/2899475.2899496>.
- Wille, K., Dumke, R.R., Wille, C., 2016. Measuring the accessibility based on Web Content accessibility Guidelines. In: Proceedings of the 2016 Joint Conf. of the International Workshop on Software Measurement and the International Conf. on Software Process and Product Measurement (IWSM-MENSURA). IEEE, pp. 164–169.
- World Wide Web Consortium, 2019a. W3C - Web Accessibility Initiative (WAI), Making the Web Accessible . URL: <http://www.w3.org/WAI/>.
- World Wide Web Consortium, 2019b. About w3C, Groups. URL: <https://www.w3.org/Consortium/activities>.
- World Wide Web Consortium, 2019c. Web content accessibility guidelines (WCAG) 2.1. URL: <https://www.w3.org/TR/WCAG21/>.
- Zaki, M., Forbrig, P., 2011. User-oriented accessibility patterns for smart environments. In: International Conf. on Human-Computer Interaction. Springer, pp. 319–327.
- Zhang, M., Wang, C., Bu, J., Yu, Z., Zhou, Y., Chen, C., 2015. A sampling method based on URL clustering for fast web accessibility evaluation. *Front. Inf. Technol. Electron. Eng.* 16 (6), 449–456.



Débora Maria Barroso Paiva received a PhD degree in Computer Science from the Universidade de São Paulo, Brazil, in 2007. Since 2006 she has been a lecturer and researcher in the Universidade Federal de Mato Grosso do Sul, Brazil. Her current research interests include software engineering, software accessibility, agile development, and human computer interaction.



André Pimenta Freire received a PhD degree in Computer Science from the University of York, England, in 2013. Since 2012 he has been a lecturer and researcher in the Universidade Federal de Lavras, Brazil. His current research interests include the areas of software engineering, inclusive design, information systems and software accessibility.



Renata Pontin de Mattos Fortes received a PhD degree in Physics from the Universidade de São Paulo, Brazil, in 1996. Since 1987 she has been a lecturer and researcher in the Universidade de São Paulo, Brazil. Her current research interests include the areas of web engineering, open source development, software accessibility, and human computer interaction.