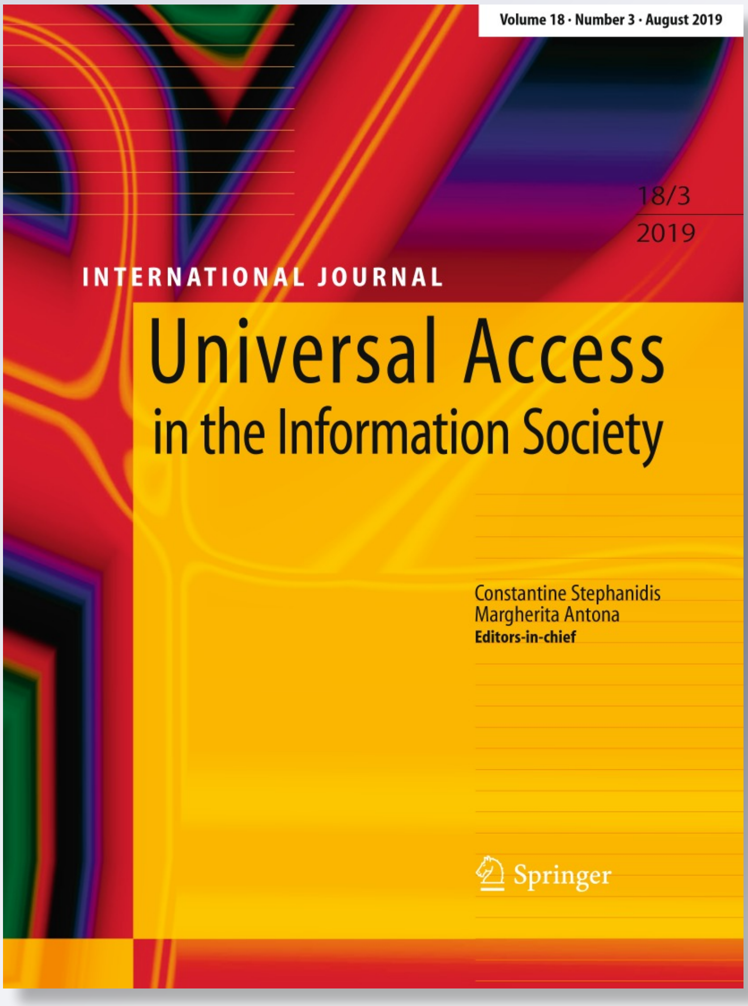
Accessible software development: a conceptual model proposal

João Sousa e Silva, Ramiro Gonçalves, Frederico Branco, António Pereira, Manuel Au-Yong-Oliveira & José Martins

Universal Access in the Information Society
International Journal

ISSN 1615-5289
Volume 18
Number 3

Univ Access Inf Soc (2019) 18:703-716
DOI 10.1007/s10209-019-00688-5



Universal Access in the Information Society (2019) 18:703–716
https://doi.org/10.1007/s10209-019-00688-5

LONG PAPER

Accessible software development: a conceptual model proposal

João Sousa e Silva[1] · Ramiro Gonçalves[1,5] · Frederico Branco[1] · António Pereira[2] · Manuel Au-Yong-Oliveira[3] · José Martins[1,4]

Published online: 28 July 2019

Abstract

Equal access to all software and digital content should be a reality in the Digital Era. This argument is something defended both by existing regulations, norms and standards, and also business organizations and governments. Despite this acknowledgement, the reality is still far from the desired equality. For certain groups of disabled or impaired citizens, such as the visually impaired, the existence of e-accessibility compliance represents an opportunity to integrate, in a more simple and straightforward manner, their societies. Despite the existing poor results on e-accessibility compliance, the mentioned citizens insist on using digital devices in their daily lives. Even though, in the last decade, multiple standards and regulations have been published towards indicating how to develop accessible digital user interfaces, there are still two major issues surrounding its implementation: the complexity and disparity of the documents containing the abovementioned norms, and also the lack of e-accessibility know-how by software experts. With this in mind, a proposal for an accessible software development model that encompasses e-accessibility incorporation as one of the development process activities has been presented. This model might represent a very interesting support tool for software development organizations and a novel resource for learning and training institutions to be able to improve their computer science and informatics students' skills on e-accessibility.

Keywords Software engineering · Software accessibility · Software development model · Accessible software

1 Introduction

Assuming that a functioning digital society is able to ensure equal access to all inherent (digital) devices, content and applications [1, 2], it should be clear that when developing software and digital content there must be a continuous effort to make it as accessible as possible [3, 4] to all intended parties. However, and despite the various efforts to shed light on the issue of digital accessibility [5, 6], much remains to be done [7]. The current situation is thus failing to oblige with existing international regulations, such as the EU Accessibility Act [8] that states that all citizens must be allowed to access Web platforms and content in the same manner, regardless of their impairments and incapacities.

The visually impaired are, by default, one of the communities of citizens (digital users) that stand to benefit from the use of digital devices and applications that help them during their daily tasks [9]. According to existing research [7, 10],

✉ Ramiro Gonçalves
ramiro@utad.pt

João Sousa e Silva
joao.sousa.silva@gmail.com

Frederico Branco
fbranco@utad.pt

António Pereira
antonio.pereira@ipleiria.pt

Manuel Au-Yong-Oliveira
mao@ua.pt

José Martins
jmartins@utad.pt

[1] INESC TEC, University of Trás-os-Montes and Alto Douro, Quinta de Prados, 5000-801 Vila Real, Portugal

[2] Instituto Politécnico de Leiria, R. Gen. Norton de Matos, 2411-901 Leiria, Portugal

[3] GOVCOPP, Department of Economics, Management, Industrial Engineering and Tourism, University of Aveiro, Aveiro, Portugal

[4] Polytechnique Institute of Bragança - EsACT, Campus do Cruzeiro, Avenida 25 de Abril, Lote 2, 5370-202 Mirandela, Portugal

[5] AquaValor - Centro de Valorização e Transferência da Água, Chaves, Portugal

visually impaired citizens are also avid users of computers and smartphones while performing ordinary tasks such as reading online newspapers, reading and writing e-mails and interacting with others through social media and online communication services.

Though various authors [2, 11, 12] deem the aforementioned arguments as critical and of vital importance, this topic has yet to receive the necessary attention from those who are primarily responsible for the development of accessible software: the developers and project managers. In fact, according to Bohman [11] and Klironomos et al. [13], the majority of software experts have been demonstrating a serious knowledge gap when it comes to understanding the techniques and regulations associated with e-accessibility, thus creating barriers for those with some sort of disability or impairment, as they are unable to use the inaccessible software products and content, hence limiting their ability to become equal members of society.

With this context in mind, the present research has been conducted to, drawing on a set of recommendations to be incorporated into software engineering classrooms and IT organizations, propose an accessible software development model. To achieve this goal, an analysis of both the existing scientific literature and the current international accessibility regulations and best practices has been done and merged with a focused analysis on existing software development models. Thus, the current document poses as both a pedagogical asset that can be included in computer science-related courses curricula, and as a technical tool for supporting project managers (and developers) who need to continuously develop accessible software.

In terms of structure, the present article is divided into six sections, the first being the introduction. Section 2 outlines an analysis of existing related work on the topic of software accessibility. Section 3 presents a summarized perspective on software accessibility existing norms, regulations, standards and best practices. In Sect. 4, we present a global perspective on the development of accessible software by focusing on a dual view. The contribution of the paper is presented in Sect. 5, where we describe an accessible software development model proposal. The sixth, and final, section presents the research implications and, in parallel, a set of identified limitations, the future research activities that were considered and some final considerations of the overall research project.

2 Related studies

2.1 Software accessibility background

When analysing the existing literature, one can easily perceive that, as a concept, digital accessibility is defined in slightly different terms, according to the context in which the concept is being characterized [14] and that if its importance to the public is consensual, the overall opinion on the necessity of specific training on existing regulations and guidelines is not [15].

As argued by W3C [16], "…the Web is fundamentally designed to work for all people, whatever their hardware, software, language, culture, location, or physical or mental ability. When the Web meets this goal, it is accessible to people with a diverse range of hearing, movement, sight and cognitive ability…". Despite distinguishing between accessibility and usability, W3C states that the terms overlap in several aspects. Regarding usability, W3C claims "…usability and user experience design significantly overlap with accessibility when "specified users" includes people with a range of disabilities and "specified context of use" includes accessibility considerations such as assistive technologies. However, the needs of people with disabilities are often not sufficiently addressed in usability practice and research. Additionally, accessibility includes a technical aspect that is usually not a focus of usability. In practice, basic accessibility is a prerequisite for usability…" [17].

As presented by Baptista et al. [5], the International Organization for Standardization has also presented its interpretation of the accessibility concept. ISO [18] is a multipart standard that covers several aspects of ergonomics of human–computer interaction and joins usability and accessibility. According to the said standard, usability is the "extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use". Despite focusing its attention on a full range of user characteristics and capabilities, regarding accessibility, the ISO interpretation is not strictly limited to users with a formal disability.

The perception and characterization of accessibility have also been the work focus of various entities worldwide. Examples of this work are the EU [19] standard on the accessibility requirements for public procurement of ICT products and services in Europe (en301549), and US [20] Section 508 that enforces accessibility and usability constraints to public ICTs.

The standard EN301549 was developed by the European Telecommunications Standards Institute (ETSI), as a result of Mandate 376, which covers accessibility requirements for the public procurement of products and services in the ICT domain. This regulation is intended to join, in a single source, detailed, practical and quantifiable functional accessibility requirements. It takes into consideration the global initiatives in this field, which are applicable to all ICT products and services. Hence, it is supposed to be used in public procurement and as a source of information to make conformity assessments.

Public procurement can be an important instrument to prompt accessibility. It is critical that procurers make clear and very well-defined requirements. To make this happen, EN301549 has, in “Chapter 4—Functional Performance”, and in “annex C-Determination of Compliance”, respectively, information to help procurers to make a clear definition of accessibility requirements, and tools to guide them in their assessment and evaluation of compliance levels.

The standard EN301549 is also envisioned to be a critical document to be followed by all ICT developers, considering its high level of detail, ease of access and organized structure. As stated, EN301549 considers global initiatives in this field. Therefore, ISOs and WCAG are assimilated into it.

Thus, existing knowledge on software accessibility and usability patterns and compliance criteria is not only very well developed but also fully available to all who wish to fulfil and comply with accessibility and usability requirements. From a scientific perspective, we believe that the existing research on the topic is already very advanced and additional efforts towards further extrapolating the topic would not result in significant contributions for both theory and practice.

2.2 Is accessibility considered as a real and relevant necessity?

In order to assess how the scientific community is approaching the digital accessibility issue, a direct analysis has been made of six of the most relevant international scientific repositories (Web of Science, Science Direct, IEE Xplore Digital Library, SpringerLink, Wiley Inter Science Journal Finder, and IET Digital Library), by searching their content using the following keywords as filters: “software accessibility”, “digital accessibility”, “Web accessibility” and “mobile accessibility” (Fig. 1).

By analysing these indicators, it is evident that the topic of digital accessibility, and more specifically the production and delivery of accessible software, has yet to be properly considered by the scientific and academic communities. Hence, these facts can have a major negative influence in spreading digital accessibility concerns, research and routines.

Accessible software benefits all users, not only those with disabilities or impairments [21]. Although this fact should be enough to make every developer want to produce accessible software, and every stakeholder requests it, the actual situation is still very far from this [6, 22]. As argued by Vollenwyder et al. [23], if a given software, regardless of its nature, is accessible to all users then it will not only trigger impaired users to use it but will also allow them to take advantage of all its content and information. Nevertheless, these same authors also posit that software and content creators tend to impose a personal beliefs system responsible for triggering the will to incorporate accessibility features on their creations.

According to Martins et al. [14] and Gambino et al. [24], there is a clear necessity for accessible software and, as a consequence, for technical and functional knowledge on how to incorporate the widely publicized and available accessibility requirements. As already argued within the existing literature, developers tend to lack the necessary skills to both understand and implement the needed accessibility requirements [25]. Also, and as argued by Baptista et al. [5] and Inal et al. [26], the exiting standards and guidelines on the topic tend to be very complex and difficult to understand, thus making the developers’ task even harder.

Despite the fact that the majority of existing research and practical work has been focusing on web-based software and computer software, the reality is that currently ICT users are giving particular attention to mobile-based applications that also need to be accessibility compliant [27]. Even though

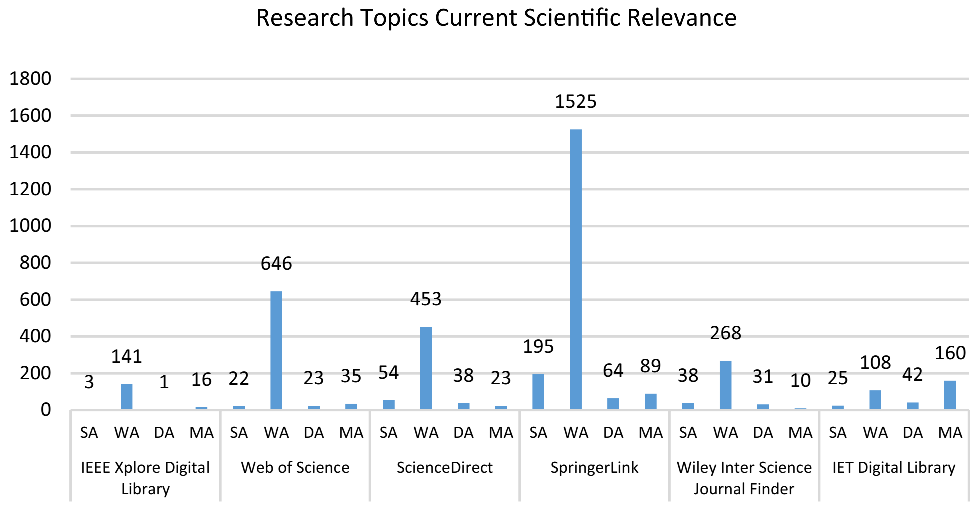


Fig. 1 Number of occurrences of “Software Accessibility” (SA), “Web Accessibility” (WA), “Digital Accessibility” (DA) and “Mobile Accessibility” (MA) as keywords in relevant scientific repositories

W3C [28] efforts towards establishing patterns aimed at the development of accessible mobile applications, according to Larco et al. [29], Eler et al. [30] and Jones et al. [31], this effort came short of reaching the desired positive results.

As argued by Draffan et al. [32] and Gonçalves et al. [33], digital accessibility is also not suitably taught, nor applied, in academic environments, and hence, the need to change mentalities and develop not only the existing theory on the topic, but also to present improvements and developments to existing practical approaches when developing software. According to Miele [34], this issue might also be related to the possible lack of knowledge of both professors and trainers on the accessibility topic, thus helping to highlight the need for a proper accessible software development model that both support those who teach and those who learn, so that they have something to lean on while performing their jobs.

2.3 Software accessibility for blind users

There have been several assessments of digital accessibility. One example, the Study on Assessing and Promoting E-Accessibility, endorsed by the European Union and published in November 2013 [35], tested e-accessibility compliance in 27 European countries and, not surprisingly, indicated that much work needs to be done in this area.

In July 2015, WebAIM conducted a survey of screen reader software from all-over the world [36], where a total of 2515 responses were validated. This survey included questions on both the screen reader software—e.g. what screen reader was being used, what operating system was being used—and the user's opinion about the evolution of Web accessibility. Also, the survey enquired about some more specific e-accessibility problems—e.g. accessibility in PDF files. However, this survey seems to assume, as its baseline, a few unclear arguments and certainties which are in fact also ignored by many stakeholders. Namely, the survey asked about Web accessibility on news websites. This may lead to the conclusion that those who work with disabled people know, empirically, that this section of the population is consumers of this type of content. Although this is a strong belief, some background work seems to be missing in order to validate this same belief, since e-accessibility is clearly not well addressed.

Digital accessibility has the potential to change the life of several groups of people with disabilities. It means that people with disabilities may have the chance to participate more meaningfully in society. However, not everybody can use the resource equally [37]. For example, visually impaired people can now, perhaps for the first time in history, read a book outside, using mainstream equipment such as a smartphone, or a tablet. Also, this group of people now has the possibility to have a portable dictionary. Continuing with the same example of visually impaired people, it is now possible to read a newspaper, accessing news at the same time as a sighted person. And, if they are using a mobile device, it is now possible to use the time in a waiting room, or seated at a cafe, to read. Simple things, like consulting the information of a product, are now reasonably easy activities, or even making a purchase using an accessible device with accessible software. Through the Internet, in an accessible manner, a blind person can manage to listen to a specific radio, or TV channel, overcoming the inaccessible devices such as set-top boxes. Several groups of people are now able to communicate, using social media, either because they are accessible by screen readers, or because it is possible to use written communication, or it is simply convenient if, for some reason, the person cannot leave the house. For a visually impaired person, it is now possible to be independent in written communication, in an accessible manner, providing him or her with another level of independence and privacy, just because they can have access to mobile services and e-mails. The facility to have a video call is simply astonishing for a deaf person, who can now use his lip-reading abilities or a gestural idiom. The possibilities are so extensive that now, a blind person can use a mobile application to call a volunteer, who, through a video call, can help the blind person, using their vision instead of the blind person. Also, it is possible to have an optical character recognition application in a smartphone that can be used, by a blind person, to check a bill, or the mail. Another possibility is to use a mobile application to overcome closed functionality equipment, such as a ticket machine, or a printer with just a tactile screen, without speech output [9, 38]. Naturally, for all of this to come to fruition, the software must be accessible.

An analysis of our initial research [7, 39, 40] showed that the majority of disabled or impaired citizens do use digital devices daily for tasks as simple as reading, working or interacting with others. From a more technological point of view, it was also possible to see that they also access the Internet from their devices using the provided Internet browsers.

A proper analysis of Fig. 2 clearly reveals that the results of one of the initially performed studies on the use of digital services highlight the importance of advocating for accessible software. If we extrapolate the survey sample used (significant set of blind and partially sighted Portuguese citizens), to an international level, one can perceive that mundane tasks, such as reading and sending e-mail messages, using social media, using search engines or even using news websites, are on the list of the most recurrent activities of those who are blind or partially sighted. Despite the importance of digital public services in order to ensure equality in the ability to reach all types of public services (social security services, tax services, etc.) [41], only a small minority of the abovementioned study participants admitted

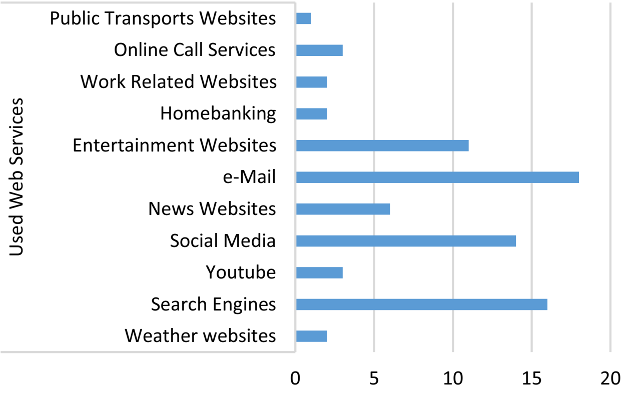


Fig. 2 How target group members use the Internet. Adapted from Silva et al. [7]

to having used digital public services to take advantage of the immense possibilities associated with them.

3 Accessible software development: existing norms, regulations and standards

Considering the existing lack of accessibility-related knowledge among developers [14] and their inability to properly develop accessible software [42], in order to spread e-accessibility awareness among them, the topic should be taught in the academic environment [34]. As shown, this topic—e-accessibility—is extremely under-addressed. According to Ko and Ladner [43], the topic is so poorly taught that it requires a baseline to even start the learning process. This assumption has been the basis for the proposal by Palan et al. [44] on the inclusion of accessibility (and usability) as relevant topics in software development courses.

Considering not only the abovementioned but also the existing literature arguments on the topic [12], we pose that the availability of documents regarding e-accessibility and the awareness on its relation with the type of user interface they "regulate" is extremely important not only to better understand the rules of engagement but also to improve the developers' perception on how to implement and validate e-accessibility. However, despite the overwhelming number of documents regarding e-accessibility, not all of them present new (and disruptive) arguments to the discussion. Hence, in Table 1, we present a relation between the various types of user interface and the documents that software developers should consult to develop accessible and usable interfaces. These documents have been selected according to their relevance and their scientific and technical recognition.

In addition to this guidance, there are some specific governmental rules, such as Sect. 508 from the USA [21], which may be consulted. However, the above recommendations overlap with these governmental guidelines. Actually, these national recommendations, such as the Brazilian eMAG [45], are in line with the international recommendations mentioned above. Although ISO 9241-171 can be used, EN301549 can be used instead of it. As mentioned, it includes global accessibility initiatives, including ISO 9241-171. The standard EN301549 has the advantage of being free of charge.

3.1 Applying existing e-accessibility patterns to develop accessible software

As software is making its way towards the cloud and "local software" is being transformed into web-based [46], the need for compliance with existing Web accessibility patterns is also growing [47]. In what concerns the referred type of applications and the inherent user interfaces, the Web Content Accessibility Guidelines (WCAG) of the World Wide Web Consortium are the recommended accessibility standard for many organizations—including governmental organizations—for the establishment of an accessible Web for people with disabilities [48]. These are comprised of guidelines and checkpoints to ensure a certain level of accessibility addressed to specific disability-related problems [23]. Nevertheless, it is important to keep in mind that Web accessibility is a very complex and broad topic. This means that not everybody has to know it fully [49].

For native applications with standard controls, the human interface guidelines from the host operating system are mandatory in case developers and want to create a graphical user interface (GUI) in line with the operating system style

Table 1 Relationship between each type of document and its appropriate type of user interface

Type of user interface (UI)	Documents to consult
Web UI	Web Content Accessibility Guidelines (WCAG) or chap.9 of EN301549
Native application with standard controls from the host operating system	Human interface guidelines from the host operating system; accessibility programming guide of the host operating system
Native application with UI controls made from scratch	Host operating system accessibility APIs; Chap. 11 of EN301549
UI for a big software system, such as an operating system and machines with closed functionality, such as ticket machines	ISO 9241-171:2008—Ergonomics of human–system interaction or EN301549

[50]. It is relevant to mention that when a developer keeps the same graphic style from the host operating system on his/her application, he/she is already increasing the level of usability (and by inherence the accessibility), since the interaction will be like the rest of the system [51]. Therefore, there is probably no need for a specific learning curve. Using standard graphical components, the developer would not have to make them particularly accessible, since they are already built with the accessibility features provided by the accessibility APIs. Consequently, using the standard components, developers would just have to consult the accessibility programming guide of the host operating system in order to use those components accurately [52, 53].

Choosing to build graphical components from scratch means that a lot of accessibility research and projection work will be wasted. Operating systems like some versions of Windows, iOS, Android, MacOS, etc. have already incorporated a lot of development work regarding accessibility [54]. When choosing not to use standard graphical components, the developer must be aware that he/she is wasting a lot of work that someone else has already done, during the platform's user interface (UI) development.

According to de Souza [55] and Silva et al. [7] for the creation of a bigger UI for a larger software system, such as an operating system, the recommendation is to use ISO 9241-171:2008—Ergonomics of human–system interaction—Part. Prepared by Technical Committee ISO/TC 159, Ergonomics, Subcommittee SC 4, Ergonomics of human–system interaction, this ISO standard "provides ergonomics guidance and specifications for the design of accessible software for use at work [20], in the home, in education and in public places", as stated in its abstract. This should be the guidance for a big, new UI, built from scratch.

4 Developing accessible software

Despite having already been acknowledged as a true necessity, not only for those with some necessities or impairments, but also to those living an ordinary life without any type of physical or psychological constraint, software accessibility is yet to be encompassed as an essential software feature for those in charge of planning, designing and developing software.

As argued by Gonçalves et al. [56] and Martins et al. [6], even though the global consensus on the relevance of accessible software and on the need for accessibility to be included as one of the software features when performing the initial analysis and specification, the truth is that the existence of a multi-perspective on the topic has led to the (negative) current state of things, where the majority is not accessible to all [57]. This same assumption has been made by Alič [58] when arguing that despite the fact that organizations are trying to ensure compliance with the ISO 9001 quality standard there is a serious difficulty in ensuring software quality and compliance with existing technical regulations and guidelines.

4.1 Organizational and government perspectives

At the end of the day, the decision to include accessibility features is always based on the customer. If a few years ago the inclusion of the referred features was still considered as a non-requirement and a topic generally unknown [22, 59], this situation has been reversed and now organizations are starting to incorporate accessibility-related knowledge in their software development processes [42, 60] due to ethical and social motives but, most of all, for a financial reasons as disabled and impaired "users" are valued in the billion dollar category [61, 62]. This, however, is not a global situation, and there are several examples of public and private organizations that use (and develop) software and digital platforms that are not accessible [63].

According to various authors [64–66], the arguments that have been highlighted earlier (lack of interest in the topic [22], assuming accessibility as an expense and not as an opportunity [67, 68], accessibility as a technicality impossible to fully implement with a modern and beautiful interface [69], irrelevance of the topic in what concerns existing regulations and no compliance-related consequences [70]) are becoming less relevant and not only business managers are considering it very important to implement software that can be accessed by all, but also governments are aiming at both improving their website accessibility compliance and, at the same time, further extending existing regulations towards enforcing both public and private organizations to implement accessibility software [71, 72].

Hence, it is clear that the acceptance barrier has, in a global manner, been eliminated though there are still multiple challenges that need to be focused on and addressed in order for accessible software to be a complete reality [73, 74].

As presented by Diament [75], IT companies have also embraced the challenge to improve their digital platforms' accessibility as they recognize that this will not only help disabled or impaired users in the "present" but will also help the overall IT users in the future given that the incorporation of accessibility features tends to also improve the usability of the referred platforms by those without any type of impairment.

After further analysis of the issues concerning the development of accessible software, IT companies all agree that major efforts must be made not only to increase their teams' focus on ensuring that produced software are accessible to all users, including those with some sort of disability [76], but also to improve their employees' skills on accessibility and usability [75, 77].

According to Taylor [78], the existing lack of knowledge on the usability and accessibility topics has led business managers to assume, at some extent, that these are extremely complicated topics that require extensive knowledge and expertise. This has led to the analysis, planning, development and continuous ensuring of accessibility and usability compliance to IT teams where most of the members are also not educated on both the technologies and techniques for developing accessible and usable platforms and, most of all, for creating accessible and usable content. Aiming on addressing these issues, there are various entities performing valuable work such as for example the National Federation of the Blind Jernigan Institute that has published an introductory guide for SMEs and startups to start incorporating accessibility features in their organizations' software-related activities [79].

4.2 Accessible software quality assurance

In order to ensure that systemic software projects solve problems, firms have been adopting software quality assurance strategies and methodologies supported on recurrent validation tasks where the newly developed software components are assessed against previously defined functional and non-functional requirements [80]. This assurance of software quality levels throughout its lifecycle is extremely important considering that currently software is in permanent evolution [81].

As argued by Sánchez-Gordón, Moreno [82], despite the known relevance of the software accessibility topic, it is extremely important to acknowledge the necessity of not only incorporating it in software quality assurance processes and procedures, but also to understand how developers can simply merge their knowledge on how to use the accessibility tools and features of their software development tools, and the accessibility requirements that have been imposed early on in the project. In order to complement developers' compliance with existing accessibility and usability standards, software development teams need to have experts on testing not only the main software features but most importantly its conformity with existing regulations, standards and guidelines [83]. As argued by Freire et al. [84], testing software is both important for traditional software and for Web-based applications, and hence the need to ensure developers' knowledge on the software accessibility topic is broader and allows for an accurate programming of all types of software.

Despite the previously enumerated (and described) most prominent and globally accepted standards, regulations and guidelines, when one analyses the software quality paradigm it is impossible to avoid both the ISO/EIC 9126 software quality model [85] and the ISO/IEC 25000 technical norm [86]. The ISO/IEC 25000 norm highlights the existence of eight main software quality features: (1) Functional Suitability; (2) Performance Efficiency; (3) Compatibility; (4) Usability; (5) Reliability; (6) Security; (7) Maintainability; (8) Portability. According to ISO, in order for a given software piece to be able to be considered truly usable, it must ensure an appropriate recognizability, a good level of both learnability and operability, it has to ensure user error protection, present user interfaces with a good level of aesthetics and also be accessible to all users including those with some sort of disability or incapacity [87].

Hence, as one can easily understand from the abovementioned, software developers must not only understand the usability and accessibility guidelines, but also possess the necessary knowledge on how to test their developments against existing usability and accessibility standards and guidelines, thus ensuring their end-product is of the upmost quality.

4.3 Accessibility experts' perspective

Despite the current state of things, where digital and software accessibility is now starting to be seen by organizations and governments as a very relevant topic, there has been a fair share of scientific research and technical work made with the focus on not only developing methods and techniques for developing and accessing accessible software, but also on developing (and continuously improving) tools for performing the necessary developments and consequent compliance evaluations [5].

As argued by Silva et al. [7] and Peixoto et al. [70], accessibility experts acknowledge the abovementioned matter as something that has to be perceived with a dual-approach: improving and normalizing existing regulations and guidelines and, in parallel, improving software engineers' and software developers' skills on the accessibility and usability themes by upgrading their courses' curricula.

In what concerns existing regulations, accessibility experts argue that the amount of existing (national and international) regulations and guidelines makes it extremely difficult to acquire the necessary knowledge to implement accessible software. On top of this, the existing normative on the topic is usually confusing and full of extremely specific technical jargon that makes its interpretation, and consequent implementation and validation, very difficult [62].

From an education and training perspective, and as argued by Davis [88] and La Rocca [89], software developers still lack the ability to use existing tools for developing accessible software and for testing its compliance with existing regulations. Even if the current offer for accessibility and usability quick-training courses and specializations is very interesting, traditional computer science and informatics courses do not include this topic as an element of their curriculum [5].

5 An accessible software development model proposal

Considering our best knowledge on the existing scientific research, opinion articles and technical reports on the software accessibility topic, we do perceive that this is something that is still not fully integrated in software development tasks because software developers and project managers cannot support their work on a development model that encompasses the accessibility analysis, planning and implementation. With this in mind, an effort has been made to analyse existing software development models and to create a proposal for an accessible software development model that perceives the accessibility issue as something that is mandatory during the analysis and specification stages and the development and testing stages.

5.1 Conceptual model proposal: overall perspective

Software development process models are essential for the proper control and management of software development projects, and these vary depending on the organizations and the nature of the projects in the number of phases, in the order of their execution, in the tools and methods employed, on the controls and products that need to be delivered and on an endless amount of more or less important details. The specification of software development process models allows to define generic processes. The structural activities are applied to any project; however, the tasks to be performed in each activity must be adjusted according to their individual characteristics. Figure 3 presents the conceptual model of the software development process that aims to propose the inclusion of accessibility throughout the software life cycle.

Drawing on a hybrid methodology, the presented model combines both user-centred and agile development methodologies ("UCASD—User-Centred Agile Software Development") [90] and encompasses six major phases that should be sequentially performed.

As argued by Larson, Chang [91], the principles behind agile software development are "…individuals and interactions over processes and tools; working software over comprehensive documentation; customer collaboration over contract negotiation; and responding to change over following a plan. The result of following these ideals, software development becomes less formal, more dynamic, and customer focused…". According to Albadarneh et al. [92], despite the possible risk associated with the agile approach, it tends to diminish the risk of presenting software solutions that are inadequate or that do not meet the necessary functional and technical requirements. The development of accessible software is directly related with the existence of a software development process that allows for a continuous evolution of both the software and user interfaces [39], thus mimicking the current business environment and organizations' changing needs [56, 93].

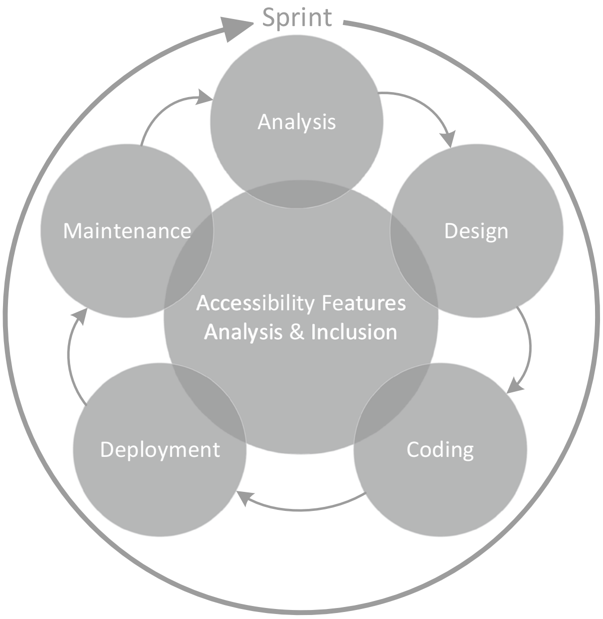


Fig. 3 Software development process model

Even though the conceptualization of user-centred design lacks global consensus, existing research still presents a series of principles that tend to be considered of general acceptance [94]: user focus, active user involvement, evolutionary systems development, simple design representations, prototyping, evaluate use in context, explicit and conscious design activities, a professional attitude with multidisciplinary teams, including usability experts, holistic design, process customization and establishing a user-centred attitude.

Thus, combining both agile and user-centred approaches makes sense, given the relevance of allowing for software analysts, software developers, software testers, user interface designers and software "owners", to actively and continuously collaborate in the pursuit of including accessibility concerns and patterns in software development projects [90].

5.2 Conceptual model stages

The proposed model consists of five phases. Each phase includes activities centred on tasks dedicated to accessibility. We describe them below.

5.2.1 Analysis

As argued by Sommerville and Sawyer [95], in order to produce a software solution that at the same time is complete and complies with all system, functional, business and standard-related requirements, it is indispensable to perform a focused and detailed analysis of both the software solution surrounding it and the impairing needs that it should fulfil. In order to properly design a software solution to a given problem, one needs to perform a focused investigation of all the details of the existing problem [96]. Thus, in order for a development team to start designing their software solution, they need to accurately identify all existing functional and non-functional requirements.

In our conceptual model, the analysis phase is responsible for system-level requirements gathering activities. This is a high-level analysis focused on strategic information planning and business area analysis. This phase is crucial to decide the feasibility of the project, identifying the need to develop a new system or the improvement of an existing one. It should be delimited in its scope and a base pane elaborated where the time and resources necessary are declared for its execution. Subsequently, the software requirements must be specified in conjunction with the client, considering the domains of information, function, behaviour, performance and interfaces. For each type of user interface, the analyst should make a merge effort between the e-accessibility standards (available in Table 1) and the functional and non-functional requirements of the software to be developed. According to Breaux et al. [97], when software analysts accurately identify and characterize the accessibility (and usability) requirements of a given software solution and its relation with all the other functional and non-functional requirements, than the chances of successfully reaching an accessible final product increase.

5.2.2 Design

According to Wieringa et al. [96], after reaching the list of requirements that must be fulfilled, software architects and UI experts might combine efforts towards designing a solution that is not only appealing and easy to use but also complies with all identified necessities.

The design phase proposed in our conceptual model should represent the translation of software requirements into a set of representations to define the model in which it will be developed, focusing on the specification of data structure, software architecture, algorithms and the characterization of interfaces. Software projects tend to be divided into logical and physical components, the logical component being an abstraction of the computational platform and the physical component aligned with the technological details inherent to the coding choices (e.g. databases, programming languages, hardware, operating system and data communications). When idealizing and designing user interaction interfaces, the UI design experts should assume the need for usability and accessibility requirements, thus idealizing user-friendly interfaces for all. Drawing on the arguments of Kuzma [98], software designers tend to lack the necessary knowledge and experience on how to design accessible and usable user interfaces that all users, regardless of their inabilities, would be able to use.

5.2.3 Coding

As already mentioned, software developers tend to lack the skill set to develop usable and accessible software that meets existing e-accessibility and usability norms and regulations. Even though coding tends to be an individual activity where each programmer uses his/her own coding approach and organization, the existing (technical and scientific) literature stipulates the guidelines for both coding according to existing standards but also to always be critical towards developers' work as one can in most of the cases improve not only the efficiency but also the code compliance of the mentioned patterns [22, 99].

As we propose in our accessible software development conceptual model, the coding stage encompasses the creation of computer programs based on the project specifications, that is, the translation of the project into programming languages, the creation of databases and the implementation of the system support platform. The greater the design detail, the greater the automation in code generation. When coding the applications—based on the previous specification—the plugins and tools that the IDEs (Integrated Development Environments) already incorporate must be used, not only for the development of accessible software, but also for the immediate evaluation of compliance.

Hence, by combining the abovementioned with Kim [100] and Martins et al. [14], it is critical that software development teams possess, at least, the basic knowledge on how to incorporate the accessibility features in their coding activities and, in parallel, on how to properly test the developed software "modules" against existing norms or standards. The referred teams should also consider the need for the developed software to have a high level of maintainability, thus making it easy to ensure a continuous fulfilment of all requirements (including those related with accessibility and usability) [101].

5.2.4 Deployment

According to Dearle [102], in traditional software development models, the deployment stage is known to be a post-production stage where the software piece is made available to a "production" server, i.e. for it to be publicly available.

The high complexity behind this stage makes it critical for a more complete project success and is very dependent on multiple factors that might influence the stage outcome, such as the necessity for a specific configuration in the software hosting server [103].

In the proposed conceptual model, this stage assumes a determinant role given that when deploying the software, it must be ensured that all accessibility and usability features are active and fully functional. This is relevant at this stage because some of the features may require specific configuration requirements by platform hosting servers. After the deployment tasks are performed, software testers should revise the entire software solution in order to discard possible accessibility or usability issues. In case, testers detect some anomaly, it should be immediately reported to both the project manager and the deployment manager in charge.

5.2.5 Maintenance

In their research, Riaz et al. [104] argue that in order for a given software to maintain its overall functionality and usability, it must have a significant level of maintainability. On a similar note, Agarwal, Majumdar [105] also hold that ensuring a considerable maintainability will trigger the software's ability to continuously oblige with existing standards such as the usability and accessibility norms, regulations and guidelines. Given the nature of software, it will inevitably have to evolve. Changes arise from the emergence of new errors, from the adaptation to changes in the external environment, from the prevention of future problems or from continuous functional improvement. Over time, it should be ensured that the content to be created is in line with existing regulations and that compliance must be periodically verified through the use of automated assessment tools and verified less regularly through the use of real users (with or without a disability).

Our perspective on the maintenance stage is that it should be considered something more than simply correcting errors and issues that have been identified. Those who are responsible for performing the maintenance tasks should also help those who analyse, design and develop the software by giving them feedback on their efforts for compliance with existing e-accessibility and usability norms, and on the possibilities of evolution that might be considered for future developments.

6 Conclusions

The e-accessibility issue, despite being the focus of more research today than a few years ago, has yet to be the target of a study that summarizes and condensates existing documentation (norms, rules, regulations, standards and guidelines) in order for those analysing, developing and testing software to be able to acquire know-how in a single place.

From a theoretical perspective, merging into a single document a set of acknowledgements from the existing literature on e-accessibility and merging this with not only the motives behind the importance of the topic for both science and practice, but also with a conceptual model that allows software developers and software development project managers to create accessible software, is a very considerable contribution.

As mentioned above and highlighted, despite the relevance associated with the e-accessibility topic and the existence of multiple research projects focused on the topic, the presentation of the motives behind organizations' and governments' willingness to adopt e-accessibility and the combination of these motives with experts' perspectives on the topic might serve as the basis for future research aimed at not only further developing the techniques, methods and tools to develop accessible software, but also to further develop existing norms.

The presentation of a proposal for an accessible software development model is, from our perspective, a novel contribution that might represent something of tremendous value for those IT companies willing to develop accessible software, as it encompasses the accessibility feature (and inherent activities) into the software development process. This was possible as our research team includes individuals who have the benefit of both academic (and therefore theoretical) as well as practical experience in software development and implementation. Furthermore, academia also tends to be more socially aware and less profit-conscious, an aspect which led to the writing of this particular research study. From yet another perspective, a conceptual software development model with the ability to support the development of accessible software might serve as the basis for learning and training institutions to adapt the way in which they teach students about how to develop software and, as such, thus starting to teach how to develop accessible software. This would address one of the most highlighted motives for the existence of non-accessible software, the lack of know-how, in the professional marketplace, on how to do it.

The presented work is a part of a broader research project in which a new (and fully tested and matured) software development model is to be developed. This said one of the limitations of the present work is that it presents a conceptual model that is yet to be included in a real test, such as a software development project with real developers and real requirements. Thus, as future work, we are going to plan and execute a real-life test where a given piece of software is to be developed with the structural support of our accessible software development model. With this test, we aim to not only test the integration between all of the defined development stages, but also assess the level of compliance with accessibility regulations and guidelines of the produced software piece.

Digital accessibility is a very under-implemented feature. Those with decision-making power are not likely to care or even know enough about it. Not enough attention has been given to the subject in the *media*, and making the required alterations, to make software accessible to all, is still very much seen more as a cost than as an opportunity to widen one's customer base. Since software accessibility is not being fulfilled, and its power is under-evaluated, a possible and reasonable contribution to the global solution to the highlighted issues would be to propose an accessible software development model that incorporates the accessibility topic in its core. This conceptual model could be used not only for supporting software development projects aimed at developing accessible software, but also as a basis for software engineering and development teaching and training activities, thus helping to reduce the gap between existing accessibility regulations, guidelines, tools and evaluation procedures and computer science and informatics students and IT specialists.

Acknowledgements This work was supported by national funds through the Portuguese Foundation for Science and Technology (FCT) under the Project UID/CEC/04524/2016. This work was financed by the ERDF—European Regional Development Fund through the Operational Program for Competitiveness and Internationalization—COMPETE 2020 within Project (POCI-01-0145-FEDER-006961), and by National Funds through the Portuguese funding agency, FCT—Fundação para a Ciência e a Tecnologia as part of Project (UID/EEA/50014/2013).

References

1. Pick, J., Azari, R.: Global digital divide: influence of socioeconomic, governmental, and accessibility factors on information technology. Inf. Technol. Dev. **14**(2), 91–115 (2008)
2. Persson, H., Åhman, H., Yngling, A., Gulliksen, J.: Universal design, inclusive design, accessible design, design for all: different concepts—one goal? On the concept of accessibility—historical, methodological and philosophical aspects. Univ. Access Inf. Soc. **14**(4), 505–526 (2015)
3. Woody, M., Lewis, J., Greenberg, M., Goldman, Y., Ostap, E.: Accessible, feature-rich software for rigorous model fitting using maximum likelihood estimation. Biophys. J. **110**(3), 331a (2016)
4. Desruelle, H., Isenberg, S., Botsikas, A., Vergori, P., Gielen, F.: Accessible user interface support for multi-device ubiquitous applications: architectural modifiability considerations. Univ. Access Inf. Soc. **15**(1), 5–19 (2016)
5. Baptista, A., Martins, J., Gonçalves, R., Branco, F., Rocha, T.: Web accessibility challenges and perspectives: a systematic literature review. In: 2016 11th Iberian Conference on Information Systems and Technologies (CISTI), Las Palmas, Spain, 15–18 June 2016, pp. 1–6. IEEE (2016). https://doi.org/10.1109/cisti.2016.7521619
6. Martins, J., Gonçalves, R., Branco, F., Pereira, J., Peixoto, C., Rocha, T.: How ill is online health care? An overview on the Iberia Peninsula health care institutions websites accessibility levels. In: New Advances in Information Systems and Technologies, pp 391–400. Springer, Berlin, (2016)
7. Silva, J., Gonçalves, R., Martins, J., Pereira, A.: Making software accessible, but not assistive: a proposal for a first insight for students. In: World Conference on Information Systems and Technologies, pp. 149–156. Springer, Berlin (2017)
8. EU: European Accessibility Act. vol COM/2015/0615 final—2015/0278 (COD). Employment, Social Affairs & Inclusion (2018)
9. Ashraf, M., Hasan, N., Lewis, L., Hasan, M., Ray, P.: A systematic literature review of the application of information communication technology for visually impaired people. Int. J. Disabil. Manag. **11**, E6 (2016). https://doi.org/10.1017/idm.2016.6
10. Okonji, P., Lhussier, M., Bailey, C., Cattan, M.: Internet use: perceptions and experiences of visually impaired older adults. J. Soc. Incl. **6**(1), 120–145 (2015)
11. Bohman, P.: Teaching Accessibility and Design-for-All in the Information and Communication Technology Curriculum: Three Case Studies of Universities in the United States, England, and Austria. Utah State University, Utah (2012)
12. Putnam, C., Dahman, M., Rose, E., Cheng, J., Bradford, G.: Best practices for teaching accessibility in university classrooms: cultivating awareness, understanding, and appreciation for diverse users. ACM Trans. Access. Comput. (TACCESS) **8**(4), 13 (2016)
13. Klironomos, I., Antona, M., Basdekis, I., Stephanidis, C.: White paper: promoting design for all and e-accessibility in Europe. Univ. Access Inf. Soc. **5**(1), 105–119 (2006). https://doi.org/10.1007/s10209-006-0021-4
14. Martins, J., Gonçalves, R., Branco, F.: A full scope web accessibility evaluation procedure proposal based on Iberian eHealth accessibility compliance. Comput. Hum. Behav. **73**, 676–684 (2017). https://doi.org/10.1016/j.chb.2016.12.010
15. Yesilada, Y., Brajnik, G., Vigo, M., Harper, S.: Exploring perceptions of web accessibility: a survey approach. Behav. Inf. Technol. **34**(2), 119–134 (2015). https://doi.org/10.1080/0144929X.2013.848238
16. W3C (2016) Accessibility. Standards—Web Design and Applications
17. W3C (2016) Accessibility, Usability, and Inclusion: Related Aspects of a Web for All. W3C—Web Accessibility Initiative. https://www.w3.org/WAI/intro/usable. Accessed 01 Mar 2017
18. ISO (2010) ISO 9241-210:2010—Ergonomics of human–system interaction—Part 210: Human-centred design for interactive systems. International Organization for Standardization
19. EU: EN 301 549 v1.1.1—Accessibility requirements suitable for public procurement of ICT products and services in Europe. vol EN301-549. ETSI-CEN-CENELEC (2014)
20. US.: Section 508. Reabilitation Act. US Government (2017)
21. Díaz-Bossini, J., Moreno, L.: Accessibility to mobile interfaces for older people. Proc. Comput. Sci. **27**, 57–66 (2014)
22. Gonçalves, R., Martins, J., Pereira, J., Oliveira, M., Ferreira, J.: Accessibility levels of Portuguese enterprise websites: equal opportunities for all? Behav. Inf. Technol. **31**(7), 659–677 (2012)
23. Vollenwyder, B., Iten, G., Brühlmann, F., Opwis, K., Mekler, E.: Salient beliefs influencing the intention to consider web accessibility. Comput. Hum. Behav. **92**, 352–360 (2019). https://doi.org/10.1016/j.chb.2018.11.016
24. Gambino, O., Pirrone, R., Di Giorgio, F.: Accessibility of the Italian institutional web pages: a survey on the compliance of the Italian public administration web pages to the Stanca Act and its 22 technical requirements for web accessibility. Univ. Access Inf. Soc. **15**(2), 305–312 (2016)
25. Branco, R., Cagnin, M., Paiva, D.: AccTrace: accessibility in phases of requirements engineering, design, and coding software. In: 2014 14th International Conference on Computational

Science and Its Applications, 30 June-3 July 2014, pp. 225–228. (2014) https://doi.org/10.1109/iccsa.2014.51

26. Inal, Y., Rızvanoğlu, K., Yesilada, Y.: Web accessibility in Turkey: awareness, understanding and practices of user experience professionals. Universal Access in the Information Society. https://doi.org/10.1007/s10209-017-0603-3 (2017)
27. Serra, L., Carvalho, L., Ferreira, L., Vaz, J., Freire, A.: Accessibility evaluation of e-government mobile applications in Brazil. Proc. Comput. Sci. **67**, 348–357 (2015)
28. W3C.: Mobile Accessibility at W3C. W3C WAI—Standards/Guidelines (2017)
29. Larco, A., Yanez, C., Montenegro, C., Luján-Mora, S.: Moving beyond limitations: evaluating the quality of android Apps in Spanish for people with disability. In: Cham, 2018. Proceedings of the International Conference on Information Technology and Systems (ICITS 2018), pp. 640–649. Springer, Berlin (2018)
30. Eler, M., Rojas, J., Ge, Y., Fraser, G.: Automated accessibility testing of mobile Apps. In: 2018 IEEE 11th International Conference on Software Testing, Verification and Validation (ICST), Västerås, pp. 116–126. IEEE, Sweden (2018)
31. Jones, M., Morris, J., Deruyter, F.: Mobile healthcare and people with disabilities: current state and future needs. Int. J. Environ. Res. Publ. Health **15**(3), 515 (2018)
32. Draffan, E., Wald, M., Dickens, K., Zimmermann, G., Kelle, S., Miesenberger, K., Petz, A.: Stepwise approach to accessible MOOC development. Stud. Health Technol. Inf. **217**, 227 (2015)
33. Gonçalves, R., Martins, J., Pereira, J., Santos, V., Cota, M.: Can i access my school website? Auditing accessibility of the Portuguese teaching institutions websites. J. Univ. Comput. Sci. **19**(18), 2639–2655 (2013)
34. Miele, J.: Teaching teachers and making makers: what the maker movement can teach the world about accessibility and design. In: Proceedings of the 2017 Conference on Interaction Design and Children, pp. 5–6. ACM, Stanford, California, USA (2017)
35. Kubitschke, L., Cullen, K.: Study on Assessing and Promoting e-Accessibility. EUR-OP (2013)
36. WebAIM.: Screen Reader User Survey #6 Results, WebAIM—Web Accessibility in Mind (2015)
37. Rughiniş, C., Rughiniş, R.: 'In My Shoes' interaction sandbox for a quest of accessible design: teaching sighted students accessible design for blind people. In: International Conference on Universal Access in Human–Computer Interaction, pp. 64–74. Springer, Berlin (2014)
38. Brock, A., Truillet, P., Oriola, B., Picard, D., Jouffrais, C.: Interactivity improves usability of geographic maps for visually impaired people. Hum. Comput. Interact. **30**(2), 156–194 (2015)
39. Silva, J., Gonçalves, R., Martins, J., Branco, F., Pereira, A.: Accessibility in software engineering: pursuing the mainstream from a classroom. In: Cham, 2018. Learning and Collaboration Technologies. Learning and Teaching, pp. 505–517. Springer, Berlin (2018)
40. Silva, J., Gonçalves, R., Pereira, A.: Accessibility in the software life cycle a maieutic exercise in software engineering. In: 2017 12th Iberian Conference on Information Systems and Technologies (CISTI), Lisbon, Portugal, 21-24 June 2017, pp. 1–5. IEEE. (2017). https://doi.org/10.23919/cisti.2017.7975710
41. Kamoun, F., Almourad, M.: Accessibility as an integral factor in e-government web site evaluation: the case of Dubai e-government. Inf. Technol. People **27**(2), 208–228 (2014)
42. Luís, C., Rocha, Á., Marcelino, M.: Acessibilidade em ambientes virtuais de aprendizagem: accessibility in virtual learning environments. RISTI-Revista Ibérica de Sistemas e Tecnologias de Informação **25**, 54–65 (2018)
43. Ko, A., Ladner, R.: Access Computing promotes teaching accessibility. ACM Inroads **7**(4), 65–68 (2016)
44. Palan, N., Hanson, V., Huenerfauth, M., Ludi, S.: Teaching inclusive thinking in undergraduate computing. In: Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility, pp. 399–400. ACM, (2017)
45. eMAG.: Modelo de Acessibilidade em Governo Eletrônico, vol. v3.1. D. d. G. Eletrônico (2005)
46. Lanich, Z.: The Benefits of Moving to The Cloud. Forbes (2017)
47. Giraud, S., Thérouanne, P., Steiner, D.: Web accessibility: filtering redundant and irrelevant information improves website usability for blind users. Int. J. Hum. Comput. Stud. **111**, 23–35 (2018)
48. Santarosa, L., Conforto, D., Machado, R.: Whiteboard: synchronism, accessibility, protagonism and collective authorship for human diversity on Web 2.0. Comput. Hum. Behav. **31**, 591–601 (2014)
49. Putnam, C., Wozniak, K., Zefeldt, M., Cheng, J., Caputo, M., Duffield, C.: How do professionals who create computing technologies consider accessibility? In: Proceedings of the 14th International ACM SIGACCESS Conference on Computers and Accessibility, pp. 87–94. ACM (2012)
50. Inostroza, R., Rusu, C., Roncagliolo, S., Rusu, V.: Usability heuristics for touchscreen-based mobile devices: update. In: Proceedings of the 2013 Chilean Conference on Human–Computer Interaction, pp. 24–29. ACM (2013)
51. Corbett, E., Weber, A.: What can I say?: addressing user experience challenges of a mobile voice user interface for accessibility. In: Proceedings of the 18th International Conference on Human–Computer Interaction with Mobile Devices and Services, pp. 72–82. ACM (2016)
52. Obrenovic, Z., Abascal, J., Starcevic, D.: Universal accessibility as a multimodal design issue. Commun. ACM **50**(5), 83–88 (2007)
53. Park, K., Goh, T., So, H.: Toward accessible mobile application design: developing mobile application accessibility guidelines for people with visual impairment. In: Proceedings of HCI Korea, 2014. Hanbit Media, Inc., pp. 31–38
54. CEUD.: Irish National IT Accessibility Guidelines. vol 1.1—Ensure that users have access to the operating system accessibility tools, without affecting application functionality. Centre for Excellence in Universal Design (2014)
55. de Souza, E.: Acessibilidade web: diferentes definições e sua relação com o design universal. Diálogo com a Economia Criativa **1**(1), 13–28 (2016)
56. Gonçalves, R., Rocha, T., Martins, J., Branco, F., Au-Yong-Oliveira, M.: Evaluation of e-commerce websites accessibility and usability: an e-commerce platform analysis with the inclusion of blind users. Univ. Access Inf. Soc. **17**(3), 567–583 (2018). https://doi.org/10.1007/s10209-017-0557-5
57. Schiavone, A., Paternò, F.: An extensible environment for guideline-based accessibility evaluation of dynamic Web applications. Univ. Access Inf. Soc. **14**(1), 111–132 (2015)
58. Alič, M.: Integration of the ISO 9001 QMS with the company's IT business system. Total Qual. Manag. Bus. Excell. **29**(9–10), 1143–1160 (2018). https://doi.org/10.1080/14783363.2018.1487216
59. Abanumy, A., Al-Badi, A., Mayhew, P.: e-Government Website accessibility: in-depth evaluation of Saudi Arabia and Oman. Electron. J. e-Gov. **3**(3), 99–106 (2005)
60. Ismailova, R.: Web site accessibility, usability and security: a survey of government web sites in Kyrgyz Republic. Univ. Access Inf. Soc. **16**(1), 257–264 (2017)
61. Bhawalkar, G., Truog, D., McClean, C., Nickels, A., Neuburg, S.: The billion-customer opportunity: digital accessibility. Forrester (2018). Retieved from https://www.forrester.com/report/The+BillionCustomer+Opportunity+Digital+Accessibility/-/E-RES143294#

62. Leitner, M., Strauss, C., Stummer, C.: Web accessibility implementation in private sector organizations: motivations and business impact. Univ. Access Inf. Soc. **15**(2), 249–260 (2016). https://doi.org/10.1007/s10209-014-0380-1
63. Acosta-Vargas, P., Acosta, T., Luján-Mora, S.: Challenges to assess accessibility in higher education websites: a comparative study of Latin America Universities. IEEE Access **6**, 36500–36508 (2018). https://doi.org/10.1109/ACCESS.2018.2848978
64. Rybarczyk, Y., Jadán, J., Villarreal, S., Esparza, W., Acosta-Vargas, P., Guevara, C., Nunes, I.: Analysis and improvement of the web accessibility of a tele-rehabilitation platform for hip arthroplasty patients. In: Advances in Human Factors and Systems Interaction: Proceedings of the AHFE 2018 International Conference on Human Factors and Systems Interaction, July 21–25, 2018, Loews Sapphire Falls Resort at Universal Studios, Orlando, Florida, USA, p. 233. Springer, Berlin (2018)
65. Archambault, D., Dupire, J.: Digital games accessibility. In: Cham, Computers Helping People with Special Needs, pp. 241–244. Springer, Berlin (2018)
66. Rozado, D., Haden, P.: Otago polytechnic accessibility software hub: an open source repository of accessibility software for motor impairment. In: Paper Presented at the Proceedings of the 29th Australian Conference on Computer–Human Interaction, Brisbane, Queensland, Australia (2017)
67. Groves, K.: How Expensive is Web Accessibility? (2011). http://www.karlgroves.com/2011/11/30/how-expensive-is-accessibility/. Accessed 12 May 2018
68. NCDAE: Indicators for Institutional Web Accessibility (2014). http://www.ncdae.org/goals/indicators.php#b3c. Accessed 15 Jun 2018
69. Farrelly, G.: Practitioner barriers to diffusion and implementation of web accessibility. Technol. Disabil. **23**(4), 223–232 (2011)
70. Peixoto, C., Branco, F., Martins, J., Gonçalves, R.: A multi-perspective theoretical analysis to web accessibility. In: Anabela, M. (Ed.) Research Paradigms and Contemporary Perspectives on Human–Technology Interaction, pp. 117–139. IGI Global, Hershey (2017). https://doi.org/10.4018/978-1-5225-1868-6.ch006
71. Ghosh, S., Srivastava, A.: Accessibility and facilitating e-Governance through open educational resource movement. Educ. Quest **8**(3), 781–788 (2017)
72. Juárez-Ramírez, R.: User-centered design and adaptive systems: toward improving usability and accessibility. Univ. Access Inf. Soc. **16**(2), 361–363 (2017). https://doi.org/10.1007/s10209-016-0480-1
73. Takagi, H., Asakawa, C.: New challenges in web accessibility. Univ. Access Inf. Soc. **16**(1), 1–2 (2017). https://doi.org/10.1007/s10209-015-0436-x
74. Sanchez-Gordon, S., Sánchez-Gordón, M., Yilmaz, M., O'Connor, R.: Integration of accessibility design patterns with the software implementation process of ISO/IEC 29110. J. Softw. Evol. Process **31**(1), e1987 (2018). https://doi.org/10.1002/smr.1987
75. Diament, M.: Tech giants team up to boost accessibility. Disabilityscoop (2015). Retieved from https://www.disabilityscoop.com/2015/07/23/tech-team-boost-accessibility/20495/
76. D'Onfro, J.: How Google has stepped up its efforts to make its own tech more accessible to the disabled. CNBC (2018). Retrieved from https://www.msn.com/en-us/money/markets/how-google-has-stepped-up-its-efforts-to-makes-its-tech-more-accessible-to-the-disabled/ar-BBM5KsG
77. Mura, A.: Why accessibility is a vital part of SaaS innovation. Mediumcom (2017). Retrieved from https://medium.com/@Userlane/why-accessibility-is-a-vital-part-of-saas-innovation-af8f52d782bb
78. Taylor, Z.: Web accessibility: not just for tech experts anymore. Disabil. Compliance High. Educ. **23**(9), 5 (2018). https://doi.org/10.1002/dhe.30416
79. NFJI.: Digital accessibility: an introductory guide for small businesses and startups. In: Institute NfotBJ (ed) (2018)
80. Dawn, M., Deepak, K.: Software quality assurance. In: Terry, T.K. (Ed.) Handbook of Research on Technology Project Management, Planning, and Operations, pp. 242–260. IGI Global, Hershey (2009) https://doi.org/10.4018/978-1-60566-400-2.ch016
81. Mall, R.: Fundamentals of Software Engineering. In: 5th Edition Edn. PHI Learning Pvt. Ltd., Delhi (2018)
82. Sánchez-Gordón, M., Moreno, L.: Toward an integration of web accessibility into testing processes. Proc. Comput. Sci. **27**, 281–291 (2014). https://doi.org/10.1016/j.procs.2014.02.031
83. Saldaña-Ramos, J., Sanz-Esteban, A., García-Guzmán, J., Amescua, A.: Design of a competence model for testing teams. IET Softw. **6**(5), 405–415 (2012)
84. Freire, A., Goularte, R., de Mattos Fortes, R.: Techniques for developing more accessible web applications: a survey towards a process classification. In: Proceedings of the 25th Annual ACM International Conference on Design of Communication, pp. 162–169. ACM (2007)
85. Jung, H., Kim, S., Chung, C.: Measuring software product quality: a survey of ISO/IEC 9126. IEEE Softw. **5**, 88–92 (2004)
86. Koh, S., Jiang, J.: What should using a software product and usability of the software product be? J. Inf. Technol. Appl. Manag. **24**(3), 73–92 (2017)
87. IS.: ISO 25000—Software product quality. The ISO/IEC 25000 series of standards, vol 25000 (2018)
88. Davis, A.: Most developers fail to design websites accessible to people with disabilities. The Institute, The IEEE news source (2017)
89. La Rocca, D.: Seventy percent of websites are breaking the law on accessibility—here's how and why that needs to change. The Huffington Post (2017). Retrieved from https://www.huffingtonpost.co.uk/damiano-la-rocca/website-accessibility_b_9931304.html
90. Brhel, M., Meth, H., Maedche, A., Werder, K.: Exploring principles of user-centered agile software development: a literature review. Inf. Softw. Technol. **61**, 163–181 (2015). https://doi.org/10.1016/j.infsof.2015.01.004
91. Larson, D., Chang, V.: A review and future direction of agile, business intelligence, analytics and data science. Int. J. Inf. Manag. **36**(5), 700–710 (2016). https://doi.org/10.1016/j.ijinfomgt.2016.04.013
92. Albadarneh, A., Albadarneh, I., Qusef, A.: Risk management in Agile software development: a comparative study. In: 2015 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), 3–5 November 2015, pp. 1–6. https://doi.org/10.1109/aeect.2015.7360573 (2015)
93. Highsmith, J., Cockburn, A.: Agile software development: the business of innovation. Computer **34**(9), 120–127 (2001). https://doi.org/10.1109/2.947100
94. Gulliksen, J., Göransson, B., Boivie, I., Blomkvist, S., Persson, J., Cajander, Å.: Key principles for user-centred systems design. Behav. Inf. Technol. **22**(6), 397–409 (2003). https://doi.org/10.1080/01449290310001624329
95. Sommerville, I., Sawyer, P.: Requirements engineering: a good practice guide. Wiley, Hoboken (1997)
96. Wieringa, R., Maiden, N., Mead, N., Rolland, C.: Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. Requir. Eng. **11**(1), 102–107 (2006)
97. Breaux, T., Antón, A., Boucher, K., Dorfman, M.: Legal requirements, compliance and practice: an industry case study in

accessibility. In: International Requirements Engineering, RE'08. 16th IEEE, pp. 43–52. IEEE (2008)

98. Kuzma, J.: Accessibility design issues with UK e-government sites. Gov. Inf. Q. **27**(2), 141–146 (2010). https://doi.org/10.1016/j.giq.2009.10.004
99. Tsui, F., Karam, O., Bernal, B.: Essentials of Software Engineering, 4th edn. Jones & Bartlett Learning, USA (2016)
100. Kim, G.: Human–Computer Interaction: Fundamentals and Practice, 1st edn. Auerbach Publications, New York (2015)
101. Hegedűs, P., Kádár, I., Ferenc, R., Gyimóthy, T.: Empirical evaluation of software maintainability based on a manually validated refactoring dataset. Inf. Softw. Technol. **95**, 313–327 (2018)
102. Dearle, D.: Software deployment, past, present and future. In: Future of Software Engineering (FOSE'07), Minneapolis, 23–25 May 2007, pp. 269–284. IEEE, (2007) https://doi.org/10.1109/fose.2007.20
103. Lawrance, J., Jung, S., Wiseman, C.: Git on the cloud in the classroom. In: Proceeding of the 44th ACM Technical Symposium on Computer Science Education, pp. 639–644. ACM (2013)
104. Riaz, M., Mendes, E., Tempero, E.: A systematic review of software maintainability prediction and metrics. In: Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement, pp. 367–377. IEEE Computer Society (2009)
105. Agarwal, M., Majumdar, R.: Software maintainability and usability in agile environment. Int. J. Comput. Appl. **68**(4), 30–36 (2013)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.