

# Conducting Acceptance Tests for Elderly People on the Web

## Using the GPII Preference Set for a Personalized Evaluation

Alexander Henka<sup>1</sup>(✉), Andreas Stiegler<sup>1</sup>, Gottfried Zimmermann<sup>1</sup>,  
and Thomas Ertl<sup>2</sup>

<sup>1</sup> Stuttgart Media University, Stuttgart, Germany  
{henka, stiegler, gzimmermann}@hdm-stuttgart.de

<sup>2</sup> Institute for Visualization and Interactive Systems (VIS),  
University of Stuttgart, Stuttgart, Germany  
thomas.ertl@vis.uni-stuttgart.de

**Abstract.** Due to the overlapping requirements with people with disabilities, elders can benefit from accessible web design and the use of assistive technologies. But elderly face also semantic problems that are derived from different perception models or the mere anxiety of using new technologies, which can't be evaluated by accessibility guideline conformance only. Tackling those semantic issues calls for more user-centered evaluation. The Global Public Inclusive Infrastructure (GPII) provides user interface adaptation based on peoples individual needs and preferences. These preferences are stored in so-called preference sets and can also contain semantic settings. In this paper, we propose an accessibility evaluation method, using the preference sets of the GPII to derive authentic accessibility requirements. Hereby, we're able to carry out tests according to guideline conformance and semantic requirements. In this context, we propose a personalized accessibility evaluation approach based on original user preferences that addresses the need for a user-centered evaluation.

**Keywords:** Human computer interaction · Accessibility · Elderlies · Acceptance tests · Web accessibility guidelines · GPII · User-preference set · User-centered accessibility evaluation · Technical accessibility · Semantical accessibility

## 1 Introduction

The accessibility requirements of web applications for elderly people are similar to the requirements of people with disabilities. Typical issues are related to vision, cognition or motor coordination constraints [1–3]. Therefore, accessibility guidelines (e.g., WCAG 2.0 [18]) can be used as one approach to evaluate the accessibility of a web application for elderly people, as it was indicated in [2, 4, 5], but are not sufficient, as suggested by Affonso et al. [2]. They advocate an extension to the WCAG 2.0 to meet specific accessibility requirements of elderly people. Based on their research, Pernice, Estes and Nielsen [19] have generated a set of guidelines that is specifically tailored to “senior citizens” on the Web.

Elderly users however, are frequently confronted with issues that cannot be identified by merely checking the conformance of an application to guidelines. These issues result from mental models that differ between designers and users, and from differences in strategies for the meaning making and problem solving process [6]. Those difficulties are not considered accessibility related issues in the general understanding of accessibility evaluation, according to guideline conformance, but need to be tackled in the application's design [2, 4]. Age-related disability issues also tend to change over time and new or additional impairments can arise [4, 7]. This calls for a constant tracking and adaptation of accessibility requirements for ICT products.

Studies [8–11] advocate for a strong User Centered Design approach, in which testing for web accessibility can benefit from existing work in usability evaluation. Furthermore, they suggest that accessibility in general is strongly user-centered. Barriers occur when there is a mismatch between a user and a web application in terms of interaction characteristics. Brajnik [8] characterizes accessibility in a user-centered context: “[Accessibility is, when] specific users with specific disabilities can use it [the software] to achieve specific goals with the same effectiveness, safety and security as non-disabled people”. According to this understanding, a barrier is a condition, which prevents a specific user, who has specific traits and is using specific assistive technologies, from achieving their specific goals.

Hence, a barrier is an attribute of the interaction between the user and the system, and the occurrence depends therefore on the user, the situation and the system. A barrier is defined by: The user and their specific traits, the assistive technologies being used, the goal(s) of the user and the properties (defects) of the web application, which, taken together, prevent the accomplishment of the goals. In consequence, conformance tests are not sufficient to fully clarify the accessibility of a web application, as discussed in [12, 13]. The conformance to accessibility guidelines is a technical property of the website and does not take the specific traits of the user, their devices and assistive technologies into account. Hence, when checking the accessibility of a web application, it is important to take the perspective of the end user, i.e. in a way that reflects how an end user would interact with the system [10].

It was stated that especially the elderly often encounter barriers on the web, which can't be determined by guideline conformance [2, 4, 6]. These issues originate mostly from the field of bad application design or usability flaws [2, 4]. To achieve accessibility, especially for elderly people, one must not only guarantee guideline conformance, but also the usability and understandability of a web application; therefore, we have to consider technical and semantical accessibility.

**Technical Accessibility.** Describes the conformance of an ICT product to accessibility guidelines. Even if guideline conformance is accomplished, there is no guarantee that this will result in an accessible product. Accordingly, accessibility-testing tools, which rely on guideline conformance only, can't ensure the accessibility of a product, if web applications are just designed to pass the conformance test [22, 23].

As the Web Accessibility Initiative (WAI) phrases it: “[...] *Web accessibility evaluation tools can not determine the accessibility of Web sites, they can only assist in doing so.*” [21]. It is relatively easy for an automatic tool to determine if an “alt” attribute is present for an image, but hard to tell if the alternative text conveys any meaning to a

specific user. But nevertheless, technical accessibility is a requirement, since the occurrence of a barrier is also connected with defects in the web application [8].

**Semantical Accessibility.** Characterizes that the content must be understandable and usable for the user. A web application could be perfectly accessible according to guideline conformance, but can still be inaccessible to some users. If one doesn't understand how to proceed at a certain stage of "purchasing a product" in an online shop, it is a barrier [8]. Strategies to overcome those issues are widely known in the school of usability design. On prominent example is the concept of personas, as introduced by Cooper in [24]. Personas are illustrations of hypothetical users. They are composed by an aggregation of stories, requirements, needs, goals and preferences of the product's end users. Personas are used to clarify features or the workflow of an application by rising questions like: "Would Anna know what to do at this point of the purchasing process?" or "Is it clear for Anna what to enter in this web form?" (Where Anna is the name of a persona) [25].

According to [19] elderly often have anxieties in using systems, which they are not familiar with. Keeping the interaction paradigm consistent across application of the same kind empowers users to apply their learned interaction patterns they are familiar with. It is crucial to know who the products audience is, what their motivations, skills, goals, and preferences are, in order to meet the semantic accessibility.

The remainder of this paper is structured as follows: Sect. 2 introduces the general concept of our approach. Section 3 provides an introduction to the preference set of the *Global Public Inclusive Infrastructure* (GPII) [31] and how we envision deriving accessibility requirements. Section 4 provides a discussion on our work and an outlook on the current status and the next steps.

## 2 Concept of Persona-Based Accessibility Testing

Testing the accessibility is of special importance for *Rich Internet Applications* (RIAs) [26]. Typically those web applications have a complex behavior and generate large parts of their user interface components at runtime. Therefore, the accessibility of a RIA cannot be fully clarified by looking at the HTML code only, since lots of user interface elements are generated at runtime due to user interaction. This demands the conduction of acceptance tests [17], and involves testing the application as rendered in the browser, with dynamic code (e.g., JavaScript) being executed, and the user interacting with the application.

We are proposing an accessibility evaluation method for RIAs, focusing on technical and semantic accessibility. We understand conducting acceptance tests as testing a web application in the same way an end user would interact with. Our evaluation method is semi-automatic, which means that we are testing for technical accessibility, based on the WCAG 2.0 conformance, but also for semantic accessibility, by pointing the tester to potential problems derived from the end users' requirements and preferences. To control and simulate user interaction in a browser we use *Selenium* [32], an open source framework that is capable of simulating user interaction (e.g., interacting with buttons or filling in web forms) in the browser.

To provide a user-centered evaluation approach, we adapt the *persona* concept and use it as a vehicle to aggregate information about what and how a certain web application should be tested [14]. The typical domain of a *persona* is the actor's role in a scenario. This consists of a setting, the initial situation, and a storyline, the plot. The exact process of how a scenario evolves depends on the actor, their decisions, and traits. If the actor is exchanged, the scenario can have complete different outcome, despite the same initial situation and plot [27]. In order to use personas to conduct accessibility tests, we have to work in their domain.

So, rather than walking a list of technical problems that has been generated by a conformance-testing tool, the developer evaluates an application's accessibility based on the application-specific *use cases* and *user scenarios*. We define a user scenario as an instance of a *use case*, but with concrete usage data. For each use case of a web application, a user scenario with concrete usage data can be produced, by describing an interaction pattern for a dedicated use case. To underline its reusable character, we call this a *Blueprint* - since the role of the actor is exchangeable. This Blueprint consists of real usage data and an interaction pattern.

Each user scenario has an actor, typically represented by a *persona*. A persona consists of stories to illustrate the success criteria of accessible guidelines. Therefore, development teams can empathize with their target audience and use techniques like the barrier walkthrough to question features and functions.

Personas also provide a machine-readable representation, which consists of a persona-specific interaction pattern, such as keyboard-only navigation, and a set of relevant WCAG 2.0 [18] success criteria. The Blueprints and the personas provide the input for the acceptance tests. They tell the test system what and how to test and which interaction patterns to use, e.g., "check only for specific success criteria (e.g., WCAG 2.0) that are relevant for a particular preference set (persona), and consider navigation by keyboard or screen reader".

These tests are executed in the web browser of the client, as a simulation of how end users interact with the application. The basic principle of the system is to follow the navigation pattern, according to the Blueprint, but with the accessibility constraints imposed by the persona. Thus, the test system can automatically perform end-user acceptance tests, as introduced in a similar approach [17].

In extension to [14], we now use the GPII preference set [20] as a foundation to derive the user-specific interaction patterns and accessibility constraints for our machine-readable personas. Since a user can adjust their settings any time, and for any application, to fit their current needs, we are able to track the changing accessibility requirements of age-related disability issues, as discussed in [4]. The concept of this approach is illustrated in Fig. 1.

## 2.1 Blueprint Description

We designed a domain-specific language to describe the nature of a Blueprint, based on the work of Watanabe, Fortes and Dias [17]. They introduce a language to describe the interaction of people with disabilities in acceptance tests. They use explicit statements to express a specific navigation pattern, e.g.; use tab navigation. In contrast to

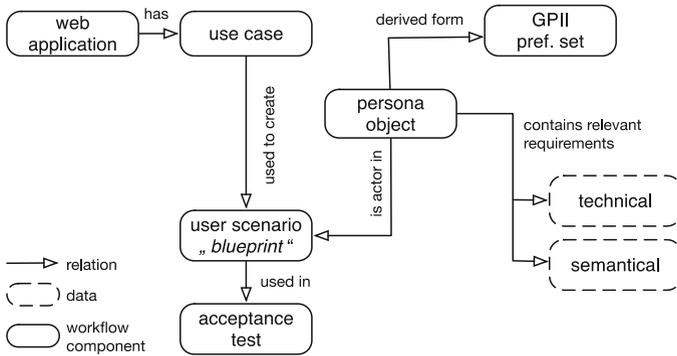


Fig. 1. Overview of the concept and the accessibility evaluation process

Watanabe, Fortes and Dias, a Blueprint doesn't include any specific interaction or navigation instructions. It defines only the interaction intention with certain elements, but not how this interaction, e.g., by mouse or keyboard, is performed, since the interaction is specific to a dedicated actor; just like the individual accessibility requirements. As the name implies, a Blueprint is an interaction draft, including only the setting and the plot. The outcome, also known as the level of accessibility relating to an application's use case, depends on the derived accessibility requirements and could be different amongst actors. Listing 1 shows an example of a Blueprint description.

The test system performs live interaction in the web browser. For every step in the Blueprint, the system performs an accessibility check, which is done automatically for those cases that can be clearly judged programmatically. For the cases where human judgment is required, e.g., verification for an image's alternative text, the system should hold the execution of the Blueprint and present the potential issues to the human tester. The tester can directly inspect the issues and decide about its validity. The same process applies for issues that fall under semantic requirements. By using GPII preference sets that are specific to an application, we can provide an evaluation method that is specific for this application, based on the application's end user requirements.

Since the test runs semi-automatically, the tester is part of the interaction and can empathize with the accessibility requirements more directly. Ignelzi [28] illustrated it nicely by using the metaphor of a farm for describing our knowledge and awareness for a certain problem domain. Everybody lives on their own farm and our knowledge and experience to a particular problem domain is limited to what we know and experienced on our own farm. By visiting other people's farms and seeing the same situation through their eyes, our understanding and awareness for a specific problem domain increases. This implies that if one experiences the interaction, issues, and problems others stumble upon; one can get a better understanding for accessibility requirements. Changing the point of view can be an asset for web authors to make accessibility guidelines more comprehensible [15, 16]. Furthermore, the use of personas is a nice way to organize such trips to farms of other people [24, 25, 29, 30].

**Listing 1.** Illustration of a Blueprint on basis of a use-case: “product search” in a fictitious web shop.

**Scenario:** Product Search

**Given** The actor navigates to: "www.WebShopExample.com"

**When** The actor enters "Sneakers" into the text field whose @id is "productSearchField"

**And** The actor interacts with a button labeled with "Search"

**Then** The actor should be on a web page with "Search Result - Sneakers" in the title

### 3 The GPII Preferences Set

The GPII [31] is a generic system to transfer user needs and preferences between devices and contexts. One of the core artifacts within the GPII process flow is the needs and preference set. Such a set is specific per user and stores preferences and settings for applications used in the past. When new devices or contexts are encountered, the GPII tries to reconstruct the way known applications were set up for new, unknown applications. The following example scenario illustrates this auto-adaptation process.

**Auto Adaptation Example Scenario: Martha.** Martha has a notebook for surfing the internet and writing emails. At work, she has a personal computer and writes product reviews and uses an office suite for writing texts, both at home and at work. She is glad to have managed to use this application, and does not know much about other software. When she works with her office suite, she uses the computer mouse in the office and the touchpad on her notebook at home. For other features, such as text formatting, she uses the toolbar of the application. At work she received an assignment to write a report, together with her colleagues. They have agreed to use a new internet-based tool that allows them to work in parallel on the same text. It also promises to adapt automatically to the user’s needs and preferences. Yet, the tool is different from the office application Martha is familiar with and she has never used a similar tool before.

Martha goes to the website and registers for the new tool. Before she can start using it, she has to fill out a questionnaire that asks a few questions about her preferences for different applications, such as what software she uses, whether she uses keyboard shortcuts and, if so, what kind of shortcuts.

In fact, the user interface of the web application looks very similar to her desktop program. She can start working immediately without learning the new interface. There are a few functions she does not know; however, the program informs her about the new functionality and tells her that other users (who have a similar set of preferences) have rated these functions as very practical. Martha reads the descriptions and decides that two of the four functions might be useful for her and disables the other two functions. Internally, the system recognizes the user adaptations and stores them in the user preference set, so the functions will no longer appear.

Over the time, Martha starts to use other Web applications and is very pleased about how well the user interfaces adapt to her behavior and needs. In particular, changes she makes in one application are also reflected in the other applications. Meanwhile, she wants this functionality for all of her applications.

A GPII preference set stores settings for an application for various contexts. Such a representation is sufficient for many use cases that cover a wide array of accessibility adaptations, for example increased font size, zoom factor, screen-reader settings etc. In the GPII, a context is a set of environmental information, such as lighting conditions, screen size or available software stacks. The functionalities of a needs and preference set and their adaptation is illustrated by transferring settings from one application to another, either in a direct relation manner, e.g., a setting is semantically equal in both applications, or in a subset or superset relation manner, e.g., if a setting in one application can be semantically represented by multiple settings of the other application. These relations are sufficient to represent many relations, but there are large families of adaptations for which this approach is not sufficient.

One example of an adaptation that cannot be expressed sufficiently is content adaptation. While a setting can express that the user requires alternative texts for images, it does not help an application actually to produce an alternative text. Or, if a setting expresses the user's desire of having always a search field to look up content, which is important for the elderly [19], won't empower a web application to generate a corresponding form element. This can result in inaccessible applications being unable to meet the desired preferences, not because they are not programmed to do so, but because they did not provide the relevant content to perform the adaptation.

### 3.1 Deriving Accessibility Requirements

Instead of transferring preferences to adapt a specific target application, we derive accessibility requirements and transfer the user preferences into a format to be used in accessibility acceptance tests.

Preferences, like language settings or a specific contrast ratio that can be mapped to WCAG criteria, are transferred in a direct relation manner and checked by guideline conformance. Generally, one can distinguish between different types of success criteria: Success criteria that can be evaluated programmatically, success criteria that can be detected automatically but need human verification, and success criteria that cannot be detected automatically and so need human verification in any case. Therefore, the test system should point the tester to issues where human verification is required. Since we are conducting live interaction, we are able to expand the space of success criteria that can be tested automatically. An example is the navigation pattern of a user, e.g., navigation by keyboard. We can hereby test the conformance to WCAG guideline 2.1 Keyboard Accessible.

Usually semantic requirements come in the form of textual descriptions [19] or wrapped in stories, if personas are used [24, 25]; therefore, we propose to use *textual requirement-statements*, which are presented to the tester during the Blueprint execution. An example: If the use-case “*searching for a product*” is evaluated for its accessibility, the system would show a semantic requirement, e.g., “*Make sure that the search button is labeled with “Search” or an equivalent text so that it is clear where to click*”, before the Blueprint step, where the actor should interact with a button to initiate the search, is executed; cf. Listing 1.

By approving the statement, the tester confirms that the corresponding semantic requirement is met. Single properties, or combination of properties, in a preference set, can be used to aggregate requirements for semantic accessibility. The study in [19] lists semantic requirements for elderly people, which can serve as a foundation for textual requirement statements. Another approach would be that textual requirement statements are stored in the user preference sets; therefore, a user can define their preferences directly.

**Listing 2.** Illustration of the envisioned format in JSON syntax, derived from the GPII preference set and used to create the accessibility requirements of an actor. The text content is partially from [19]. This showcase refers to a use case that is characterized by the Blueprint: “product-search”, in Listing 1.

```
"product-search":{•
  "technical":{•
    "WCAG":{•
      "automatic":[•
        "Guideline3_1.3_1_1",
        "Guideline4_1.4_1_2"•],•
      "manual":[•"Guideline1_2.1_2_3"•]•
    }
  },
  "selenium":{"navigation_pattern" : "keyboard"}
},
"semantic":{
  "text_field":["Display search as an open, visible
  field on all pages (I found it usually in
  the upper right corner) with a Search button
  to its right"],
  "button":["Make sure to label the search button with
  „Search” or an equivalent text so that it
  is clear where to click”
  ,”Make sure to describe in which format the
  search query should be entered”]
}
}•
```

### 3.2 Benefits of Storing Accessibility Requirements as GPII Preferences

Part of the GPII process of transferring settings from one context to another is clustering. The GPII exploits similarity between users [33]. Similarity is derived from clustering. This clustering can be utilized to allow automatically derive matching accessibility requirements for users. For example: Given, we have 100 user preference sets, which were found to be similar, and some of them have associated accessibility requirements, similar to listing 2. Depending on the similarity-distance between those preference sets, the accessibility requirements can be propagated and even automatically altered to match the respective users.

Sometimes people can be uncertain what settings they need to accommodate their needs or from which setting or assistive technology they could also benefit from. Therefore, the GPII can make suggestions based on clustering and the preference set similarity, as illustrated in the GPII adaptation scenario in Chap. 3. Hereby, we’re able to

derive accessibility requirements that are essential for users sharing similar preferences, but which does an individual user not necessarily considered as a requirement in first place; yet, users can benefit from the compliance with these additional requirements.

As expressed by its name, a preference set reflects the needs and desires of a user. Besides requirements for technical accessibility, preferences can also include semantic requirements. By using the information derived from the preference set, we're able to work with personal accessibility requirements and won't rely on accessibility guidelines only; therefore, addressing the need for a more user centered and personalized accessibility evaluation [8–11].

The preference set is refined every time the user states new preferences or alters exiting ones. Hereby, the GPII learns more about a particular user and can provide better adaptations. Due to this process, preference sets are always up-to-date, which enables a constant tracking of accessibility requirements. This is of special interest for elderlies where accessibility demands can change over time and new or addition disabilities can accrue [4, 7].

## 4 Discussion and Next Steps

When using the GPII to transfer settings from one application to another, mismatches can also occur, e.g., if the target application doesn't provide suitable settings for properties in the preference set or applications can have settings that share some of their semantics, but also involve side effects or constraints that are not present in both applications. When transferring preferences into accessibility requirements we will most likely run into similar mismatches.

As a potential improvement, we propose to decouple the users' needs and preference sets from its usage fashion in the GPII and define a preference set format, which contains dedicated accessibility preferences and usability needs. Therefore, minimizing the potential mismatches when transferring preference sets into use-case specific accessibility requirements. The GPII preference set allows storing preferences for specific applications, so called application-specific preferences. In a first attempt, we can adopt this concept by storing use-case specific preferences for accessibility tests, as application-specific. Hereby, we would still be able to transfer preferences into accessibility requirements, but would define a distinct preference set format within the structure of GPII.

As described in [24, 25], personas are vehicles for shipping requirements, preferences and goals of a product's target audience. On this account they are highly specific to a certain product or product domain and cannot be properly used for other products. Since we use machine-readable personas to express accessibly requirements for defined uses cases of a product, they cannot be reused for accessibility tests in different products as well, especially if different requirements apply. The more specific the requirements for certain use cases are, the more precise we can state the accessibility of those use cases and the application. This restricts the application scope of our system, to test the use cases of one defined application with the preferences of the target audience.

In the usage scenario, our approach is integrated in the development process, where a web author provides Blueprints and user preferences (of the target audience) are used

to generate machine-readable personas; however, we don't advocate substituting the usage of personas or accessibility evaluation techniques. We see our approach as an addition to established workflows and as a support for a more user centered accessibility evaluation.

This project is work in progress. So far, we have developed a prototype, which is capable of conducting accessibility evaluations on a technical level. Currently, we are enhancing our prototype to provide guidance for semantic accessibility. In a next step, we will use this prototype to conduct user tests, with web authors and students from the field of computer science, to validate the workflow and the concept for its usability among web authors and analyze the effectiveness of our approach in comparison with other accessibility evaluation methods.

**Acknowledgments.** Part of the research leading to these results has been researched within the Cloud4all project. Cloud4all is an R&D project that receives funding from the European Commission under the Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 289016. This publication reflects only the authors' views and the European Union is not liable for any use that may be made of the information contained herein.

## References

1. Becker, S.A.: A study of web usability for older adults seeking online health resources. *ACM Trans. Comput.-Hum. Interact.* **11**(4), 387–406 (2004)
2. Affonso de Lara, S.M., Watanabe, W.M., dos Santos, E.P.B., Fortes, R.P.M.: Improving WCAG for elderly web accessibility. In: *Proceedings of the 28th ACM International Conference on Design of Communication*, New York, NY, USA, pp. 175–182 (2010)
3. Dickinson, A., Arnott, J., Prior, S.: Methods for human–computer interaction research with older people. *Behav. Inf. Technol.* **26**(4), 343–352 (2007)
4. Abou-Zahra, S., Brewer, J., Arch, A.: Towards bridging the accessibility needs of people with disabilities and the ageing community. In: *Proceedings of the 2008 International Cross-Disciplinary Conference on Web Accessibility (W4A)*, New York, NY, USA, pp. 83–86 (2008)
5. Arch, A.: Web accessibility for older users: successes and opportunities (keynote). In: *Proceedings of the 2009 International Cross-Disciplinary Conference on Web Accessibility (W4A)*, New York, NY, USA, pp. 1–6 (2009)
6. Leitner, M., Subasi, Ö., Höller, N., Geven, A., Tscheligi, M.: User requirement analysis for a railway ticketing portal with emphasis on semantic accessibility for older users. In: *Proceedings of the International Cross-Disciplinary Conference on Web Accessibility (W4A)*, New York, NY, USA, pp. 114–122 (2009)
7. Sloan, D., Atkinson, M.T., Machin, C., Li, Y.: The potential of adaptive interfaces as an accessibility aid for older web users. In: *Proceedings of the International Cross Disciplinary Conference on Web Accessibility (W4A)*, New York, NY, USA, pp. 35:1–35:10 (2010)
8. Brajnik, G.: Beyond conformance: the role of accessibility evaluation methods. In: Hartmann, S., Zhou, X., Kirchberg, M. (eds.) *WISE 2008*. LNCS, vol. 5176, pp. 63–80. Springer, Heidelberg (2008)

9. Cooper, M., Sloan, D., Kelly, B., Lewthwaite, S.: A challenge to web accessibility metrics and guidelines: putting people and processes first. In: Proceedings of the International Cross-Disciplinary Conference on Web Accessibility, New York, NY, USA, pp. 20:1–20:4 (2012)
10. Petrie, H., Kheir, O.: The relationship between accessibility and usability of websites. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, New York, NY, USA, pp. 397–40 (2007)
11. Kelly, B., Sloan, D., Brown, S., Seale, J., Petrie, H., Lauke, P., Ball, S.: Accessibility 2.0: people, policies and processes. In: Proceedings of the International Cross-Disciplinary Conference on Web Accessibility (W4A), pp. 138–147. ACM, New York (2007). doi:[10.1145/1243441.1243471](https://doi.org/10.1145/1243441.1243471)
12. Brajnik, G.: Web accessibility testing: when the method is the culprit. In: Miesenberger, K., Klaus, J., Zagler, W.L., Karshmer, A.I. (eds.) ICCHP 2006. LNCS, vol. 4061, pp. 156–163. Springer, Heidelberg (2006)
13. Kelly, B., Sloan, D., Phipps, L., Petrie, H., Hamilton, F.: Forcing standardization or Ac-commodating diversity? a framework for applying the WCAG in the real world. In: Proceedings of the 2005 International Cross-Disciplinary Workshop on Web Accessibility (W4A), New York, NY, USA, pp. 46–54 (2005)
14. Henka, A., Zimmermann, G.: Persona based accessibility testing. In: Stephanidis, C. (ed.) HCI 2014, Part II. CCIS, vol. 435, pp. 226–231. Springer, Heidelberg (2014)
15. Freire, A.P., Russo, C.M., Fortes, R.P.M.: A survey on the accessibility awareness of people involved in web development projects in Brazil. In: Proceedings of the 2008 international cross-disciplinary conference on Web accessibility (W4A), New York, NY, USA, pp. 87–96 (2008)
16. Greeff, M., Kotzé, P.: A lightweight methodology to improve web accessibility. In: Proceedings of the 2009 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists, New York, NY, USA, pp. 30–39 (2009)
17. Watanabe, W.M., Fortes, R.P.M., Dias, A.L.: Using acceptance tests to validate accessibility requirements in RIA. In: Proceedings of the International Cross-Disciplinary Conference on Web Accessibility, New York, NY, USA, pp. 15:1–15 (2012)
18. W3C: Web Content Accessibility Guidelines (WCAG) 2.0, 11 December 2008. <http://www.w3.org/TR/2008/REC-WCAG20-20081211/>
19. Pernice, K., Estes, J., Nielsen, J.: Senior Citizens on the Web, 2nd edn. Nielsen Norman Group, Fremont (2013). Research Report by Nielsen Norman Group
20. Iglesias-Pérez, A., Loitsch, C., Kaklanis, N., Votis, K., Stiegler, A., Kalogirou, K., Serra-Autonell, G., Tzovaras, D., Weber, G.: Accessibility through preferences: context-aware recommender of settings. In: Stephanidis, C., Antona, M. (eds.) UAHCI 2014, Part I. LNCS, vol. 8513, pp. 224–235. Springer, Heidelberg (2014)
21. Web Accessibility Initiative: Selecting Web Accessibility Evaluation Tools (2005). <http://www.w3.org/WAI/eval/selectingtools.html>
22. Brajnik, G., Lomuscio, R.: SAMBA: a semi-automatic method for measuring barriers of accessibility. In: Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility, New York, NY, USA, pp. 43–50 (2007)
23. Vigo, M., Brown, J., Conway, V.: Benchmarking web accessibility evaluation tools: measuring the harm of sole reliance on automated tests. In: Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility, New York, NY, USA, pp. 1:1–1:10 (2013)
24. Cooper, A.: The Inmates are Running the Asylum. SAMS, Indiana (2004)
25. Adlin, T., Pruitt, J.: The Persona Lifecycle. Morgan Kaufmann, San Francisco (2006)

26. Fernandes, N., Batista, A.S., Costa, D., Duarte, C., Carricco, L.: Three web accessibility evaluation perspectives for RIA. In: Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility, p. 12 (2013)
27. Carroll, J.M.: Making Use, Scenario Based Design of Human Computer Interactions. The MIT Press, Cambridge (2000)
28. Ignelzi, M.: Meaning-making in the learning and teaching process. *New Dir. Teach. Learn.* **2000**, 5–14 (2000). doi:[10.1002/tl.8201](https://doi.org/10.1002/tl.8201)
29. Schulz, T., Skeide Fuglerud, K.: Creating personas with disabilities. In: Miesenberger, K., Karshmer, A., Penaz, P., Zagler, W. (eds.) ICCHP 2012, Part II. LNCS, vol. 7383, pp. 145–152. Springer, Heidelberg (2012)
30. Bailey, C., Pearson E.: Development and trial of an educational tool to support the accessibility evaluation process. In: Proceedings of the International Cross-Disciplinary Conference on Web Accessibility, New York, NY, USA, pp. 2:1–2:10 (2011)
31. Global Public Inclusive Infrastructure, January 2015. <http://gpii.net/>
32. Selenium browser automation framework, January 2015. <https://code.google.com/p/selenium/>
33. Loitsch, C., Stiegler, A., Strobbe, C., Tzovaras, D., Votis, K., Weber, G., Zimmerman, G.: Improving. In: Assistive Technology: From Research to Practice, Vilamoura, Portugal, pp. 1357–1365 (2013)