Universal Design 2016: Learning from the Past, Designing for the Future
H. Petrie et al. (Eds.)

doi:10.3233/978-1-61499-684-2-506

Evaluation of Accessibility Testing Methods. Which Methods Uncover What Type of Problems?

Aleksander BAI[1], Heidi Camilla MORK, Trenton SCHULZ and Kristin Skeide FUGLERUD
Norwegian Computing Center, Oslo, Norway

Abstract. We examined fourteen accessibility evaluation methods and put them into four categories based on the knowledge and resources required to perform them. We also classified the methods based on whether they evaluated technical or usable accessibility. Then, we selected four of these methods from different categories to evaluate accessibility of an e-ID solution. We looked for confusing and critical issues. Each method found unique issues and overlapping issues, but no single method found more than half of the issues. This information should help inform future accessibility evaluations of other solutions and can aid other teams in selecting methods based on their specific goals and resources.

Keywords. Accessibility, simulation kit, persona testing, WCAG

1. Introduction

Accessibility and usability evaluation is an important step in achieving universal design (UD). UD should be based on a user-centered design (UCD) process, with early focus on diverse users, and an iterative development process, including accessibility and usability evaluations [1]. Users should be involved throughout the development process and the design should be driven and refined by user-centered evaluations. The evaluations' aim is to see if the solutions will be usable and accessible for a wide range of people, including people with disabilities. Therefore, evaluations should focus on people with a range of abilities to uncover a variety of accessibility and usability problems.

In practice, developers and designers who create software applications with user interfaces (UI) don't have time or resources for frequent evaluations of the UI with people with various disabilities. Unless the team has a pool of testers that also includes people with disabilities, it might not be feasible to test the usability and the accessibility of the application with a broad group of people. However, as Hasdoğan [2] has shown, designers use their own world-view and experience when creating and evaluating their own work. They are, as Colman et al. put it, "designing for themselves" when creating new products [3]. The result may be many people that cannot use a technology.

[1] Corresponding Author, Norwegian Computing Center, Post box 114 Blindern, NO-0314, Oslo, Norway; E-mail: Aleksander.Bai@nr.no.

To help making UI more accessible, several accessibility evaluation methods have emerged as a supplement to evaluations with diverse users. The methods require different knowledge and experience, but they also uncover different types of issues. One group of methods for instance typically finds technical issues while other methods find usability issues. In this paper, we extend this categorization by including coverage of critical and confusing issues as well (Section 4).

We give a short background on the different methods used for accessibility testing in the next section. Then, we describe our approach and solution used during the accessibility evaluation (Section 3). Afterwards, we present and explain the results (Section 4) before discussing them and presenting a new way of categorizing accessibility methods (Section 5). Finally, we suggest avenues for future work (Section 6).

2. Background

There are many methods for testing of accessibility [4]. Fuglerud and Røssvoll [5] separated methods into three groups, but we have chosen to split the methods into four groups based on knowledge and resources required to use the methods: (A) testing using automatic or semi-automatic tools and guidelines, (B) Simulation kit where a wearable is used, (C) expert testing, and (D) testing with users.

Automatic and semi-automatic testing requires the least amount of time, knowledge, and resources compared to the other methods. A developer or tester can install a tool and run an evaluation or do a comparison against a checklist or guideline. These also are the easiest to incorporate into an automated test process. Simulation kits have an upfront cost to find the suitable ones and the actual cost of the kit. It requires some planning, but using a simulation kit usually requires little knowledge to use and get information back. This is different from expert testing that requires an expert with knowledge about impairments and how the impairments would affect someone's experience. One may possibly need to hire the experts. The expert testing may make use of simulation kits too. All the methods need time for planning the evaluation, doing it, and analyzing the results. The most resource intensive method is user testing, because it requires lots of resources in planning of the test, recruiting participants, running the test, analyzing the results, and coordinating all the people involved.

A universally designed solution needs to be accessible to the assistive technology and usable by as wide a range of people as possible. Paddison & Englefield [6] introduced a classification of *technical* and *usable* accessibility. *Technical accessibility* refers to whether a solution can work with assistive technology and follows guidelines for universal design. It often entails checking a solution works according to accessibility specifications or tests. Technical accessibility is necessary but not sufficient for universal design. *Usable accessibility* is how usable a solution is for all people, including people with impairments. **Table 1** lists several common accessibility testing methods, whether the methods test technical accessibility, usable accessibility, or both. In Section 5, we will expand upon these to include critical and confusing categories.

2.1. Testing Using Automatic or Semi-Automatic Tools and Guidelines

There are many tools for testing accessibility that can be executed automatically. These include modules that integrate directly into a developer environment like in NetBeans [7], aDesigner [8], or plugins for web browsers like the NoCoffee plugin. WebAIM

shows how an image looks like for a person with various visual impairments [9]. There are also software for adjusting the entire experience of using a computer by adjusting the display [10], adjusting the UI and the movement of the mouse cursor to simulate vision and dexterity impairments [11], or simulating other impairments [12]. Yet most tools only simulate visual impairments or a limited variant of other impairments.

Table 1. Methods and whether they mainly cover technical or usable accessibility.

Method	Technical	Usable
Automatic WCAG-testing	X	
Manual WCAG-testing	X	X
Manual testing with different types of assistive technologies	X	X
Manual and automatic standards compliance testing (CSS, HTML, etc.)	X	
Personas walkthrough		X
Cognitive walkthrough		X
Using simulation kit	X	X
Using accessibility guidelines	X	X
Interview/focus group		X
Questionnaire		X
User testing	X	X

For semi-automatic accessibility testing, the most well known method is manual inspection using the WCAG 2.0 guidelines [13], but there are other accessibility guidelines for apps [14] or content [15]. Studies have shown that accessibility guidelines can be hard to understand and follow and may produce variable results dependent on the experience of the evaluator. This method has not increased accessibility as much as anticipated [16]–[18].

While little knowledge is needed to use the automated tools, knowledge is required to interpret the output from the tools and understand how it affects accessibility. This is also true for guidelines or checklists, where it is important to understand the meaning and concepts behind a certain guideline or checklist item. However, being an expert on the impairment itself or an expert in the accessibility domain is not required; this distinguishes this group from expert testing.

2.2. Testing Using Simulation Kits

There are many different tools people can use to simulate some aspects of an impairment they don't have. One shouldn't believe that this comes close to the experience of being disabled. It is *not* simulating being disabled, but to gain insight into issues that people with impairments can experience when interacting with a UI. The lack of awareness about this distinction is a reason to why this approach has been highly criticized [19], [20]. Simulation kits do not replace user trials or focus groups, but can enhance accessibility and usability at an early stage of the development cycle [21].

There are several items someone can use to simulate vision impairment like the Cambridge Inclusive Design Glasses. These simulate different types of vision impairments [22]. Another is the VINE spectacles that simulate peripheral and central vision loss [23]. Blue Label Goggles [24] simulate blurry, double vision that might lead to loss of balance, poor targeting, delayed reactions, and slow judgment. The SIMVIZ simulation wearable uses a see-through display [25] to simulate vision impairments (macular

degeneration, glaucoma, diplopia, etc.). Of course, simple blindfolds simulate blindness, which can then be used in combination with screen readers.

The Cambridge Simulation Gloves [22] simulate dexterity loss or arthritis by restricting hand movement. People use the gloves for testing products and prototypes, but they can also be used to test software with a mouse or external device. A simple alternative is to use thick gloves or different layers of rubber gloves. Other tools are also emerging, like a haptic hand-tremor tool to simulate different types of tremors [26].

Testers can use earplugs or headphones (with and without music) to simulate hearing impairment. Headphones can also simulate the ability to perceive high frequency levels. Noise-cancelling headphones can filter out low-frequency levels. There are also simulation kits for locomotion and reaching or stretching impairments, including a whole body suit that simulates loss of capabilities in older adults [27].

We have not found kits to simulate cognitive impairments, but one kit attempted to simulates dyslexia by using mirrors [28].

2.3. Expert Testing

A common approach in expert testing is to do an heuristic evaluation where an expert performs an evaluation of a UI against a set of accepted accessibility heuristics or principles [29]. Manual WCAG inspection is an example. It is a simple and effective method, but it has been criticized for not being reliable in identifying problems even when the same set of heuristics were used [30]. However, recent methods try to improve the reliability shortcomings [31].

It is also possible to use a *persona walkthrough* or persona testing approach [32], where an expert simulates or playacts a persona while carrying out tasks. The more knowledge the expert playing the persona has about the challenges faced by people with the type of disability, the easier it is to do a realistic and credible evaluation of the solution. This approach is informal and quick to do, but is dependent on the selected personas and the experience of the experts. Personas can also be used in combination with scenarios to make accessibility testing more realistic [33].

A *cognitive walkthrough* is another inspection technique where users' mental process is studied to assess if a UI has enough instructions cues to support users' mental state when executing a task [34]. This requires a working prototype and every step is carefully documented and classified to determine the severity of the problem. The method is time consuming and tedious since all aspects needs to be considered.

2.4. Testing with Users

User testing involves real users and is better than any approximation of impairments or mental states [35], [36]. But it requires work up front to plan the test and recruit users. In particular, as many technical accessibility issues as possible needs to be solved before involving users. Otherwise, simple issues that could have been solved earlier may cause everybody to fail (e.g., unable to login to a service or incompatibility with a screen reader). This may result in an expensive and wasted exercise for all parties involved.

3. Approach

We selected four methods for our evaluation (**Table 2**). We focused on methods that did not require user testing, covering the other three groups (A, B, C) in Section 2. We also did an additional manual WCAG test for baseline comparison.

Table 2. Overview of methods and impairments used in the evaluations.

Method	Impairments
Simulation kit	Reduced vision Reduced dexterity
VATLab	Blindness Light sensitivity
Persona testing	Dyslexia Being old
Manual WCAG testing	Multiple

We evaluated an e-ID solution. The solution uses a software certificate or an ID card with a card reader for the authentication process. This consists of a Java client and a web front-end with around ten different interfaces with different complexity. Developers focused on accessibility as part of the development cycle and some evaluation was done of the components, but delays prevented evaluating the system as a whole. Ideally, accessibility testing happen during the entire development cycle [37].

We selected visual and physical impairments for the simulation kit. We chose the Cambridge Inclusive Design Glasses to simulate reduced vision. A person with good vision needs two glasses to simulate mild vision reduction. Three glasses will simulate a vision reduction of less than 1% of the population [38]. We used the Cambridge Inclusive Design Gloves to simulate dexterity reduction since there is a card reader involved. We discussed other impairment such as: Hearing loss using earplugs, poor feelings in fingers using latex gloves, a broken hand using a sling, and several vision problems using available tools. But we concluded that hearing loss were not relevant for the solution and the others were to some extent covered already.

We used a Virtual Assistive Technology Lab (VATLab) to test various Assistive Technology used by people who have low vision or are blind [39]. The VATLab contains two different screen readers and gives a good indication of how accessible the solution is for people with blindness. We used the built-in high contrast mode in the operating system to simulate issues for people that have light sensitivity. A checklist for evaluating web pages for screen readers was developed as a part of the VATLab project [39], and we used this checklist as a part of the evaluation.

For persona walkthrough, we developed two personas: a senior citizen and a young adult with dyslexia. For each persona, an expert acted as the persona while performing the predefined scenarios. To make it more realistic, the senior citizen persona testing was conducted with 2 layers of Cambridge glasses.

Finally, we tested using the WCAG checklist as a benchmark. There is no complete automatic tool for WCAG-testing [40]. We evaluated checkpoints manually, but used browser plugins to check certain items like color contrast.

Eight participants, ranging from beginner to expert knowledge, conducted eight evaluations. Each evaluation was done by at least two people, and the results were aggregated. All the evaluations were done on the same machine with the same setup to ensure an equal test environment. Each evaluation also had a coordinator that wrote down the issues reported by the tester, and the coordinator also made notes when diffi-

culties that were not verbally expressed were observed. Two coordinators were used through the evaluations.

Table 3. Overview of scenarios for the evaluation.

Scenario	Description
1	Login with a invalid digital certificate
2	Login with a valid digital certificate
3	Login with invalid smart card
4	Login with valid smart card, but incorrect PIN code
5	Login with valid smart card and correct PIN code

The tests used five different scenarios (**Table 3**). Participants were unaware they were given invalid certificates, invalid smart cards, or invalid PIN codes. The scenarios were executed in ascending order to avoid biasing the participants as they gradually progressed further in the login process. A short pilot was conducted before the scenarios started to verify the scenario, setup, and ordering.

4. Results

During the accessibility testing 425 issues were reported from all the participants. Even though the solution only covered a small bit of functionality, there were a total of 213 distinct issues (**Table 4**). Aside from the reduced dexterity impairment simulation, which reported the fewest problems, each method uncovered issues spread evenly between the different impairments.

Table 4. Number of issues discovered for each method (with percentage).

Method	Issues	%	Critical	%	Confusing	%
Simulation kit	58	27.2%	14	24.1%	19	32.8%
VATLab	62	29.1%	47	**75.8%**	10	16.1%
Persona testing	61	28.6%	28	45.9%	46	**75.4%**
WCAG	32	15.0%	7	21.9%	7	21.9%
Total	213		96		82	

WCAG found fewer issues than the other methods, but this is explained by: 1) The other methods contains at least two different evaluations with two different impairment simulations while WCAG only contains one evaluation. 2) WCAG has 62 possible evaluation criteria while almost all the other evaluations are without an upper limit.

We classified an issue as *critical* or *confusing*. A *critical issue* is an issue that prevents someone from continuing or completing a task (e.g., difficult to read images or text because of bad contrast or resolution). A *confusing issue* is an issue caused by confusing or missing information for the given context (e.g. not understanding the purpose of a screen or not understanding how to operate a controller). Issues can be critical *and* confusing, and this was often the case. An issue that was neither critical nor confusing was classified as *minor*.

Classifying an issue as critical, confusing, or minor is subjective; the results might be different if more or other evaluators were involved. For example, an issue discovered by reduced vision simulation *Difficult to read the e-Id cards* was classified as

critical, but another issue from the same simulation *Difficult to read tooltips* was classified as minor. One can argue that both are critical (or minor).

A high number of critical and confusing issues were discovered with most of the methods. Note that a critical issue might only be critical in the context of a given disability, (e.g., an issue like *incorrect HTML tags* may be critical for people who are blind, but may be less relevant for people with reduced dexterity). For an e-ID, we felt that all critical issues were important since a critical issue might exclude someone from using the service.

The simulation kit found fewer critical issues than VATLab and persona testing. This is because most of the issues were visual problems that were sometimes annoying, sometimes problematic, but not critical as we defined above. Over 70% of the critical issues discovered with simulation kit were also marked as confusing (e.g., *not obvious that something went wrong when entering a wrong pin on smart card reader*).

The WCAG reported few critical issues, but this was because the WCAG evaluations criteria are high level. So, a single criterion covers multiple issues. For example, Criteria 4.1.2 failed because: *It is not possible to use screen reader properly*. But this problem caused 17 critical issues in the VATLab since there is a much finer granularity when testing with screen readers. The high-level criteria covering multiple issues are why WCAG also reported fewer confusing issues.

The VATLab reported the most critical issues, and many of the issues were related to poor compatibility for screen readers. We suspect that more issues could be found if not so many critical issues had stopped us. Many issues found by VATLab were technical like *No information that the checkbox is disabled*. So the number of confusing issues discovered are second lowest next to WCAG, since technical issues prevented the tester from performing an operation which could also have been confusing.

Persona testing found the most confusing issues. This is because the personas (dyslexia and senior citizen) used in the evaluation focused on usability and understanding the context. Most of the issues reported by persona testing were directly related to the evaluator not understanding the context of a screen and what was expected from the evaluator, (e.g., *hard to understand all the options since so many possibilities*). Over 78% of the confusing issues found here were also marked as critical since it was impossible for the person to complete the task; this explains the equally high number of critical issues discovered by persona testing.

We classified an issue as *unique* for a method when that issue was only discovered when using that particular method. **Table 5** shows the number of unique critical and confusing issues for the different methods. The same pattern from **Table 4** is repeated here; VATLab and persona testing maintain their high percentage of unique issues discovered for critical and confusing issues respectively. Most of the confusing issues in the simulation kit, VATLab and WCAG are the same issues in persona testing. This is not surprising since persona testing found so many confusing issues. The same overlap applies for critical issues with VATLab and the other methods' critical issues.

Table 5. Unique critical and confusing issues (bold indicates method that found most).

Method	Critical	%	Confusing	%
Simulation kit	7	12.3%	10	24.4%
VATLab	35	**61.4%**	5	12.2%
Persona testing	14	24.6%	24	**58.5%**
WCAG	1	1.8%	2	4.9%

For the different impairments in persona testing, 44.3% of the issues were overlapping. One explanation for this is that the personas require precise instructions and well described concepts. The simulation kit method had 20.7% of its issues overlapping between the evaluations. One reason could be that impairment simulations don't overlap with each other. That is, problems found using dexterity simulation are hard to discover when simulating vision impairment. The VATLab method had 24.2% of its issues overlapping because the checklist and screen readers share much of the focus area.

Issue coverage for the methods is listed in **Table 6**. Coverage is similar for all methods except WCAG, which is lower. No method found more than half of all issues.

Table 6. Issue coverage for each method.

Method	Total coverage
Simulation kit	42.0%
VATLab	44.9%
Persona testing	44.2%
WCAG	23.2%

5. Discussion

Zimmermann and Vanderheiden [33] and Fuglerud [1] advocate for using multiple methods. But to the best of our knowledge, nobody has looked at how well these methods overlap in terms of what kind of issues the methods discover. No method was superior to the others, but the VATLab and persona testing complement each other for finding critical and confusing issues. No method got more than half of the issues, but it might be expected since the problem range is very large with many issues from six different impairments.

To our surprise we didn't improve much upon the WCAG method by using simulation kit, even though many unique problems were found. This might be because simulation kit focuses on the physical limitations, like reduced vision and dexterity, and not on confusing aspects. Yet the simulation kit method with two different types of impairments discovered more problems than a WCAG evaluation, but the evaluations took longer (around two and a half hours per participant for both impairments compared to one and a half hours per participant for WCAG). It's also possible to do a WCAG evaluation by yourself while simulation kit required two people: one person to write down issues while the other person evaluates using the kit.

We expected the simulation kit method to discover more unique critical issues, but many of the issues were overlapping with the VATLab. Yet blindness and reduced vision impairments have some overlap in their issues. It's encouraging that persona

testing uncovered many confusing problems that are hard to detect with other methods. Over a third of all issues from persona testing were related to confusing problems. This indicates that persona testing focuses on usability issues instead of technical issues.

Table 7. Coverage of critical and confusing issues for methods.

Method	Critical	Confusing
Using disability simulation kit	Low	Medium
Using accessibility guidelines	High	Low
Personas walk-through	Medium	High
Manual WCAG-testing	Low	Low

Based on the findings in Section 4, we can update **Table 1** with coverage of critical and confusing issues (**Table 7**). We defined *Low* for 0–20%, *Medium* for 21–50% and *High* for 51–100%. Using **Table 5**, we then get the ranges for each method.

We think this has great value for future accessibility testing where resources are limited or a limited number of methods can be selected. A test team can decide which method to use based on the application and their experience. We hope they will select multiple methods that complement each other, and discover technical, usability, critical, and confusing issues.

6. Conclusion

During the evaluation of accessibility testing methods, we found that a combination of methods works well to discover critical and confusing issues. There is no single method that works best for finding both critical and confusing issues; it is important to include at least two different methods that covers both the critical aspect and the confusing aspect to do a proper evaluation [1]. Finally, a promising combination of the VATLab and persona testing has shown good results in this evaluation. One should consider using a combination of methods that is similar to these.

It's worth noting that persona testing requires personnel that are well trained and familiar with the persona they are playacting. All the methods we have investigated in this study are relatively quick to do and can be run by people with different levels of expertise. One could use the information here to implement an accessibility evaluation routine in situations where it is not feasible to do a proper user trial, or as a preparation to a user trial. Since the evaluatons in this study are subjective in nature, the percentages should be interpreted with care. However, this study supports previous research in that accessibility evaluations using WCAG will likely uncover less than half of the accessibility issues found through a combination of methods [17], [18].

For future work, we would like to verify the results with real users, and investigate whether the results from a user trial corresponds with our results. As noted earlier, a simulation of impairments is not a replacement for doing user trials. We could also validate the results against other accessibility testing software, services, and products to see if the same results can be applied there.

Finally, we would like to extend our work with more methods. This involves doing more studies with different methods to discover where they place on the scale.

Acknowledgments

This research is partly funded by the FutureID project. The FutureID project was funded by the EU FP7 program (Grant agreement no: 318424).

References

[1] K. S. Fuglerud, "Inclusive design of ICT: The challenge of diversity," University of Oslo, Faculty of Humanitites, 2014.
[2] G. Hasdoğan, "The role of user models in product design for assessment of user needs," *Des. Stud.*, vol. 17, no. 1, pp. 19–33, Jan. 1996.
[3] R. Coleman, C. Lebbon, and J. Myerson, "Design and empathy," in *Inclusive Design*, Springer, 2003, pp. 478–499.
[4] F. Paz and J. A. Pow-Sang, "A Systematic Mapping Review of Usability Evaluation Methods for Software Development Process," 2016.
[5] K. S. Fuglerud and T. H. Røssvoll, "An evaluation of web-based voting usability and accessibility," *Univers. Access Inf. Soc.*, vol. 11, no. 4, pp. 359–373, 2012.
[6] C. Paddison and P. Englefield, "Applying heuristics to perform a rigorous accessibility inspection in a commercial context," in *Proceedings of the 2003 conference on Universal usability - CUU '03*, 2003, no. 73–74, p. 126.
[7] NetBeans, "Accessibility Checker." .
[8] IBM, "aDesigner," 2006. [Online]. Available: https://www-03.ibm.com/able/accessibility_services/adesigner.html. [Accessed: 19-May-2016].
[9] WebAim, "WebAIM Low Vision Simulator." .
[10] Microsoft, "Visual Impairment Simulator for Microsoft Windows," 2006. [Online]. Available: http://vis.cita.uiuc.edu/. [Accessed: 19-May-2016].
[11] P. Biswas and P. Langdon, "Towards an inclusive world-a simulation tool to design interactive electronic systems for elderly and disabled users," in *SRII Global Conference (SRII), 2011 Annual*, 2011, pp. 73–82.
[12] D. Giakoumis, N. Kaklanis, K. Votis, and D. Tzovaras, "Enabling user interface developers to experience accessibility limitations through visual, hearing, physical and cognitive impairment simulation," *Univers. Access Inf. Soc.*, vol. 13, no. 2, pp. 227–248, Jul. 2013.
[13] W3C Working Group, "Web Content Accessibility Guidelines (WCAG) 2.0," vol. 2009, no. 2009–11–23. 2008.
[14] T. Schulz, F. Gladhorn, and J. A. Sæther, "Best Practices for Creating Accessible Mobile Applications," Oslo, Norway, 2015.
[15] BBC, "BBC Future Media Accessibility Guidelines v1.10," *BBC*, 2014. [Online]. Available: http://www.bbc.co.uk/guidelines/futuremedia/accessibility/.[Accessed: 23-Jan-2014].
[16] D. Giakoumis, N. Kaklanis, K. Votis, and D. Tzovaras, "Enabling user interface developers to experience accessibility limitations through visual, hearing, physical and cognitive impairment simulation," *Univers. Access Inf. Soc.*, vol. 13, no. 2, pp. 227–248, Jul. 2013.
[17] D. Rømen and D. Svanæs, "Validating WCAG versions 1.0 and 2.0 through usability testing with disabled users," *Univers. Access Inf. Soc.*, pp. 1–11, 2011.
[18] C. Power, A. P. Freire, H. Petrie, and D. Swallow, "Guidelines are only half of the story: accessibility problems encountered by blind users on the web," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Austin, Texas, USA, pp. 433–442, 2012.
[19] A. M. Silverman, J. D. Gwinn, and L. Van Boven, "Stumbling in Their Shoes: Disability Simulations Reduce Judged Capabilities of Disabled People," *Soc. Psychol. Personal. Sci.*, vol. 6, no. 4, pp. 464–471, 2014.
[20] S. French, "Simulation Exercises in Disability Awareness Training: A Critique," *Disabil. Handicap Soc.*, vol. 7, no. 3, pp. 257–266, Jan. 1992.
[21] B. Sheryl and D. Tanis, "Disability-related Simulations: If, When, and How to Use Them in Professional Development," *Rev. Disabil. Stud.*, vol. 1, no. 2, pp. 8–18, 2004.
[22] University of Cambridge, "Inclusive Design Toolkit," 2015. [Online]. Available: http://www.inclusivedesigntoolkit.com. [Accessed: 19-May-2016].
[23] Visual Impairment North East, "The Vine Simulation Package," 2016. [Online]. Available: http://www.vinesimspecs.com/.[Accessed: 19-May-2016].

[24] Fatal Vision, "Fatal Vision Blue Label Goggle," 2016. [Online]. Available: http://fatalvision.com/fatal-vision-goggle-blue-clear.html. [Accessed: 19-May-2016].
[25] H. C. Ates, A. Fiannaca, and E. Folmer, "Immersive Simulation of Visual Impairments Using a Wearable See-through Display," in *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction - TEI '15*, 2015, pp. 225–228.
[26] G. P. P. Rosati, M. Fontana, R. Vertechy, M. Carrozzino, and M. Bergamasco, "Haptic hand-tremor simulation for enhancing empathy with disabled users," in *2013 IEEE RO-MAN*, 2013, pp. 553–558.
[27] C. Cardoso and P. J. Clarkson, "Simulation in user-centred design: helping designers to empathise with atypical users," *J. Eng. Des.*, vol. 23, no. 1, pp. 1–22, 2012.
[28] Northern California Branch of The International Dyslexia Association, "The Experience Dyslexia Kit," 2016. [Online]. Available: http://dyslexia-ncbida.org/simulations-kit/.[Accessed: 19-May-2016].
[29] J. Nielsen, "Usability inspection methods," in *Conference companion on Human factors in computing systems*, 1994, pp. 413–414.
[30] L. C. Cheng and M. Mustafa, "A Reference to Usability Inspection Methods," in *International Colloquium of Art and Design Education Research (i-CADER 2014)*, Singapore: Springer Singapore, 2015, pp. 407–419.
[31] J. Chattratichart and G. Lindgaard, "A comparative evaluation of heuristic-based usability inspection methods," in *Proceeding of the twenty-sixth annual CHI conference extended abstracts on Human factors in computing systems - CHI '08*, 2008, p. 2213.
[32] T. Schulz and K. S. Fuglerud, "Creating Personas with Disabilities," in *Computers Helping People with Special Needs*, 2012, vol. 7383, pp. 145–152.
[33] G. Zimmermann and G. Vanderheiden, "Accessible design and testing in the application development process: Considerations for an integrated approach," *Univers. Access Inf. Soc.*, vol. 7, no. 1–2, pp. 117–128, 2008.
[34] P. G. Polson, C. Lewis, J. Rieman, and C. Wharton, "Cognitive walkthroughs: a method for theory-based evaluation of user interfaces," *Int. J. Man. Mach. Stud.*, vol. 36, no. 5, pp. 741–773, May 1992.
[35] J. S. Dumas and J. C. Redish, *A practical guide to usability testing.*, Revised ed. Exeter: Intellect Books, 1999.
[36] D. J. Mayhew, "Cost justification of usability engineering for international web sites," in *Cost-Justifying Usability*, Elsevier Inc., 2005, pp. 359–383.
[37] M.-L. Sánchez-Gordón and L. Moreno, "Toward an Integration of Web Accessibility into Testing Processes," *Procedia Comput. Sci.*, vol. 27, pp. 281–291, 2014.
[38] J. Goodman-Deane, S. Waller, A.-C. Collins, and P. J. Clarkson, "Simulating vision loss: what levels of impairment are actually represented?," in *Contemporary Ergonomics and Human Factors 2013*, CRC Press, 2013, p. 347.
[39] K. S. Fuglerud, S. E. Skotkjerra, and T. Halbach, "Håndbok i testing av websider med hjelpe-middel-program-vare, Virtuell hjelpe-middellab," 2015. [Online]. Available: http://www.iktforalle.no/virtuell-hjelpemiddellab/veileder.html.[Accessed: 19-May-2016].
[40] M. Vigo, J. Brown, and V. Conway, "Benchmarking web accessibility evaluation tools," in *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility - W4A '13*, 2013, p. 1.