A Method for Accessibility Testing of Web Applications in Agile Environments

Sandra Sanchez-Gordon[1], Sergio Luján-Mora[2]

[1] Department of Informatics and Computer Science, Escuela Politécnica Nacional
Quito, Ecuador
sandra.sanchez@epn.edu.ec

[2] Department of Software and Computing Systems, University of Alicante
Alicante, Spain
sergio.lujan@ua.es

Abstract. Currently, more than one billion people live with some form of disability. A person's environment has a huge impact on the extent and impact of their disability, e.g. inaccessible environments create disability by creating barriers, while accessible environments diminish disability and enable full participation and inclusion. Web applications are often the only means available for people to access certain services or to certain information, e.g. healthcare information, public services, banking, education, and entertainment. Web accessibility is the property of a web application to support the same level of effectiveness for users with disabilities as it does for users without disabilities. In this study, we propose the use of automated tools, simulators, expert-based testing, and user-based testing in the context of a comprehensive method for accessibility testing of web applications in agile environments. The proposed method consist of five stages, as defined by the International Software Testing Qualifications Board: test planning and control; test analysis and design; test implementation and execution; evaluating exit criteria and reporting; and test closure activities. For each of these stages, the method details specific tasks to perform accessibility testing of web applications in the context of agile developments.

Keywords: Disability, Web Accessibility, WCAG, ISO/IEC 25010, Software Testing Process, ISTQB, Agile.

1 Introduction

Web applications are manifestly not accessible. This is a big problem today because web applications are often the only means available for people to access several services or to certain information, e.g. healthcare information, public services, banking, education, and entertainment. This study explains the concepts of disability, accessibility, web accessibility, and software testing. It also presents the principles of the Web Content Accessibility Guidelines (WCAG 2.0) and the software product quality characteristics of the standard ISO-25010 "Systems and software engineering – Systems and software quality requirements and evaluation – System and software quality models". Then, it includes a review of relevant published research in the field of accessibility testing of web applications. Finally, it proposes a method to perform accessibility testing of web applications in agile environments. The method consists of five stages, as proposed by ISTQB: test planning and control, test analysis and design, test implementation and execution, evaluating exit criteria and reporting, and test closure activities.

The rest of this paper is organized as follows. Section 2 presents the theoretical foundation. Section 3 describes accessibility testing tools and simulators. Section 4 describes the proposed method. Section 5 presents conclusions and future work.

2 Theoretical Foundation

2.1 Disability and Accessibility

According to the World Health Organization, disability is part of the human condition (25). Disability results from the interaction between persons with certain conditions and environmental barriers that hinder their participation in society on an equal basis with others. Hence, a disability is not an attribute of the person but depends on the barriers that persons with disabilities encounter in their day to day lives.

Currently, more than one billion people live with some form of disability. This is about 15% of the world's total population, i.e. the world´s largest minority group (25). Furthermore, the number of persons with disabilities increases appreciably when taking into account not only permanent disabilities but also people with temporary disabilities due to illnesses or accidents.

We can analyze accessibility issues from two perspectives. First, personal disabilities which are those associated to body or mental impairments of the human being that can be of birth or acquired at any point in a person's life, e.g. vision, hearing, speech, motor, cognitive, and psychosocial. Second, non-personal disabilities which are those associated to situations in the environment surrounding the human being that can occur at any point in a person's life and are usually temporary, e.g. cognitive issues due to language, religion, or cultural barriers, environmental conditions, internet availability, and technology availability. A person's environment has a huge impact on the extent of their disability. Inaccessible environments create disability by creating barriers. On the contrary, accessible environments diminish disability and enable full participation and inclusion. The International Organization for Standardization (ISO) defines accessibility as "the usability of a product, service, environment or facility by people with the widest range of capabilities" (9).

2.2 Web Accessibility

Web accessibility is the property of a web application or website to support the same level of effectiveness for people with disabilities as it does for people without disabilities (16). Besides, a web application or website that is accessible for users with different needs, skills, and situations also benefits people without disabilities (19).

The Word Wide Consortium (W3C) created the Web Accessibility Initiative (WAI) with the aim of studying the problems of accessibility in the web, develop guidelines and provide resources. The WAI is recognized as an international authority on web accessibility. In 1999, WAI published the first version of the web content accessibility guidelines WCAG 1.0. In 2008, WAI published the current version, WCAG 2.0. The WCAG defines how to make web content accessible to disabled persons. WCAG establishes four principles that give the foundation for web content accessibility: perceivable, operable, understandable, and robust. Perceivable means that the information and components of the user interface should be presented to users so they can perceive them. Operable means that the components of the user interface and navigation must be operable. Understandable means that the information and manipulation of the user interface must be understandable by the users. Robust means that content must be robust enough to be reliably interpreted by a wide variety of user agents, including browsers and assistive technology. WCAG define three levels of conformance: A (low), AA (medium) and AAA (high) (20).

2.3 Software Testing Process

Software testing is a process applied in a software development project with the goal of assuring the quality of the software product. The standard ISO/IEC 25010 "Systems and software engineering -- Systems and software quality requirements and evaluation -- System and software quality models" defines eight quality characteristics for software products: functional suitability, performance efficiency, compatibility, usability, reliability, security, maintainability, and portability. According to ISO, accessibility is a sub-characteristic of usability (8).

The International Software Testing Qualifications Board (ISTQB) defines the fundamental software testing process with five generic stages: planning and control; analysis and design; implementation

and execution; evaluating exit criteria and reporting; and test closure activities (5). Each stage involves a set of tasks and sometimes, two or more tasks need to be done in parallel, e.g. time pressure may mean that test execution starts before all tests have been designed.

2.4 Review of Published Research

To the best of our knowledge, there are few relevant published research on the field of accessibility testing of software products: (1), (7), (3), (17), (2), (26), (12), (11), (4), (10), (14), (15).

Brajnik (2) presents a taxonomy of accessibility evaluation methods, review existing methods such as WCAG, and proposes the use of automated tests, screening techniques, subjective assessments, barrier walkthroughs and user testing. Herramhof et al. (7) presents two tools to support test case management for accessibility test suites. The first one creates test suites for WCAG 2.0. The second one allows the edition of test description files using XML. Goncalves de Branco et al. (4) presents a CASE tool that allows traceability of accessibility requirements from conception to coding, giving developers useful information for the construction of accessible software products. Sanchez-Gordon and Moreno (14) presents a review of proposals to incorporate accessibility requirements and evaluation tools, including the Accessibility Development Lifecycle proposed by Microsoft. Sanchez-Gordon et al. (15) presents a proposal for developing accessible software based on ISO/IEC 29110.

3 Tools for Accessibility Testing

While accessibility testing is not fully automatable, tools can significantly assist software testers and contribute to more effective testing and debugging of web applications in agile environments. There are two types of tools for accessibility testing: validators and simulators.

3.1 Accessibility Testing with Validators

Automated accessibility validators are software applications, browser plug-ins, or online services that help determine if a web application or website meets accessibility requirements, such as WCAG 2.0. These tools are a useful resource to identify accessibility issues. They are best exploited when used by testers familiar with web accessibility. Table 1 presents a selection of some of the most popular evaluation tools among the listed by (23).

Name	Description
Accessibility Developer Tools	Adds to Chrome Developer Tools, an Accessibility audit with 17 rules and an Accessibility sidebar pane in the Elements tab that provides extra debugging information. https://chrome.google.com/webstore/detail/accessibility-developer-t/fpkknkljclfencbdbgkenhalefipecmb?hl=en
AChecker	Interactive, international, customizable, web content accessibility checker. Allows testers to create their own guidelines, and author their own accessibility checks. http://achecker.ca
Photosensiti-ve Epilepsy Analysis Tool	Identify seizure risks in web content and software. The evaluation is based on an analysis engine developed specifically for web and computer applications. http://trace.wisc.edu/PEAT
Readability Grader	Check whether a web content is easy-to-read. It generates 7 different scores. https://jellymetrics.com/readability-grader/
Tenon	It is an API which can be seamlessly integrated into an existing toolset. It identifies WCAG 2.0 issues. http://www.tenon.io
Total Validator	Includes a (X) HTML validator, an accessibility validator (WCAG), a CSS validator, a spell checker, and a broken links checker. http://www.totalvalidator.com
WAVE	Provides a visual representation of accessibility issues within a web page. http://wave.webaim.org/

Table 1. Selected accessibility evaluation tools.

3.2 Accessibility Testing with Simulators

Simulators tools are software applications, browser plug-ins, or online services that simulates how users with different kinds of visual disabilities will perceive the software. Table 2 presents of some of the most popular simulation tools among the listed by (23).

Name	Description
Accessibility Color Wheel	Simulates three kinds of color blindness and it shows the result of W3C algorithms that reveal if a color pair (text/background) to use in a web page is accessible. http://gmazzocato.altervista.org/colorwheel/wheel.php
Color Oracle	Color blindness simulator that shows in real time what people with common color vision impairments will see. http://colororacle.org
NoCoffee	Vision simulator for Chrome that is helpful for understanding the problems faced by people with slight to extreme vision problems. https://chrome.google.com/webstore/detail/nocoffee/jjeeggmbnhckmgdhmgdckeigabjfbddl

Table 2. Selected simulation tools.

3.3. Accessibility testing with experts and users with disabilities

Once the testers have used the simulators and automated testing tools, it is important that accessibility experts perform heuristic testing using standards and personas (6). Also, user-based testing is necessary. It should involve potential users with disabilities, including users with aging-related disabilities and foreign language users. These users may help to identify additional accessibility barriers that are not easily discovered by automated tools, simulators, and expert evaluation alone (21).

4 Proposed Method

In this section, we present the proposed method for accessibility testing in agile environments. The method consists of five stages, as proposed by ISTQB: test planning and control, test analysis and design, test implementation and execution, evaluating exit criteria and reporting, and test closure activities. For each of these stages, the proposed method includes different types of activities and details engineering tasks including specialized tasks to perform accessibility testing of web applications in the context of agile developments. Figure 1 shows the relationships among the five stages and the main accessibility-related tasks.

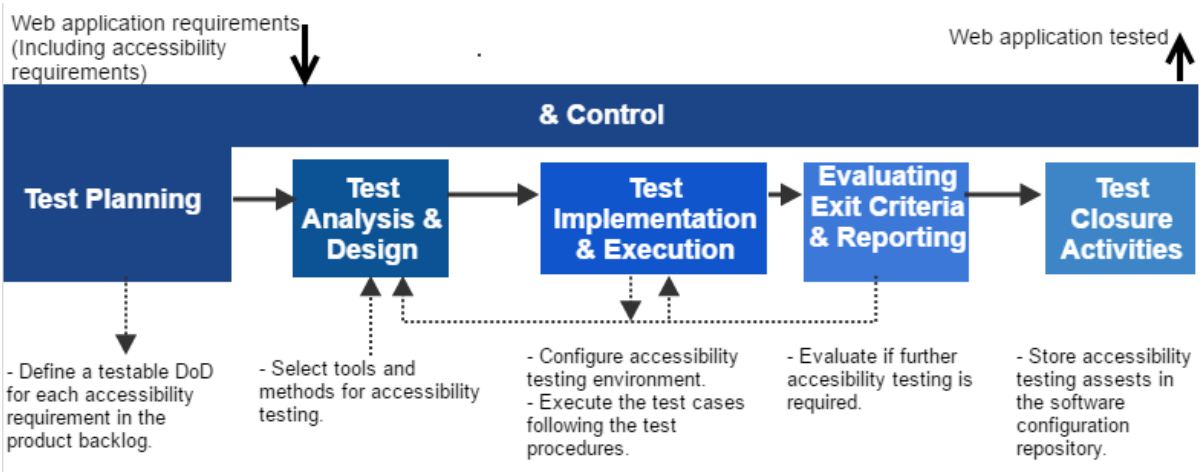

Figure 1. Method for accessibility testing in agile environments.

The initial input is the list of requirements for the web application, including the accessibility requirements, that are used in the planning part of the first stage to define a testable DoD (Definition of Done) for each item in the product backlog. The control part of the first stage will be using the testable DoD to control, monitor and evaluate the execution of the tasks of the following stages.

In the test analysis and design stage, the methods and tools for accessibility testing are selected. This second stage provides feedback for the planning and control tasks. The third stage is the test implementation and execution. In this phase the testing environment is configured and the test cases are executed following the test procedures. This stage provides feedback for the previous stage of test analysis and design. The four stage is the evaluation of the exit criteria and reporting. In this stage, findings and non-conformances are listed in the testing report. This stage provides feedback for the two previous stages: test analysis and design and test implementation and execution. Finally, the five stage is the test closure activities, that includes the storage of the testing assets for future reference and the identification of lessons learned.

4.1 Test Planning and Control

This stage prepares the agile team for the rest of stages and it selects the tools to accomplish the testing process. Table 3 shows the tasks included.

ID	TASKS	ACTIVITY
TPC1	Determine the scope of the tests, risks, objectives and strategies.	Planning
TPC2	Determine the resources of the necessary tests.	
TPC3	Implement testing strategies.	
TPC4	Create a schedule for the analysis and design of the tests.	
TPC5	Create a schedule for the implementation and execution of the tests.	
TPC6	Determine the exit criteria of the tests.	
TPC7	Measure and analyze the results.	Control
TPC8	Monitor and document the progress, coverage and exit criteria of the tests.	
TPC9	Initiate corrective actions.	
TPC10	Take decisions.	
TPC11	Sensitize the agile team through the observation of users with disabilities interacting with software products.	Accessibility
TPC12	Define a testable DoD for each accessibility requirement in the product backlog.	

Table 3. Test planning and control tasks.

4.2 Test Analysis and Design

In this stage the test environment is designed and the tools are selected. Table 4 shows the tasks included.

ID	TASKS	ACTIVITY
TAD1	Review the evidence base.	Analysis
TAD2	Identify and prioritize test conditions, test requirements or test objectives, and test data required.	
TAD3	Evaluate the testability of the requirements and the system.	
TAD4	Design specific combinations of test data, actions and expected results to cover major quality risks.	Design
TAD5	Identify the test data required for the conditions and test cases	
TAD6	Design the test environment.	
TAD7	Identify some infrastructure and some necessary tools.	
TAD8	Select accessibility evaluation tools, e.g. WAVE (24).	Accessibility
TAD9	Select HTML and CSS checkers, e.g. W3C HTML Validator (22).	
TAD11	Select simulators for different types of visual disabilities, e.g. NoCoffee.	
TAD11	Select assistive technologies, e.g. NVDA screen reader (13).	
TAD12	Select simulations aids for testing purposes, e.g. blindfolds, ear defenders.	

Table 4. Test analysis and design tasks.

4.3 Test Implementation and Execution

In this stage the test cases and procedures are developed and run. Table 5 shows the tasks included.

ID	TASKS	ACTIVITY
TIE1	Develop, implement and prioritize test cases, create test data and write test procedures.	Implementa-tion
TIE2	Prepare test harnesses and write automated test scripts.	
TIE3	Organize test sets and sequences of test procedures for the efficient execution of tests, taking into account the various constraints that could determine the order in which tests are to be performed.	
TIE4	Verify that the test environment has been successfully installed.	
TIE5	Run both manual and automated test cases.	Execution
TIE6	Record test results, including versions of the software being tested, test tools and testware.	
TIE7	Compare the actual and expected results, which may require the identification of anomalies where the actual and expected results do not match.	
TIE8	The investigation of anomalies can result in the creation of reports and the analysis of incidents.	
TIE9	Repeat the corrected or updated tests where necessary.	
TIE10	Run regression tests, when the new test versions arrive.	
TIE11	Review the design architecture, software components and interfaces for traceability with accessibility requirements.	Accessibility
TIE12	Use accessibility checklists, e.g. WebAIM´s WCAG 2.0 Checklist (24).	

Table 5. Test implementation and execution tasks.

4.4 Evaluating Exit Criteria and Reporting

The evaluation of the exit criteria and the creation of reports of the test results are strongly overlapped with the execution of the tests. Table 6 shows the tasks included.

ID	TASKS	ACTIVITY
ECR1	Check the test records against the exit criteria of the tests specified during test planning.	Evaluation of exit criteria
ECR2	Evaluate whether further testing is required or whether the specified exit criteria should be modified.	
ECR3	Write a test summary report for business stakeholders.	Reporting
ECR4	Evaluate whether further accessibility testing is required.	Accessibility

Table 6. Evaluating test criteria and reporting.

4.5 Test Closure Activities

As the execution of the tests reaches a close, the exit criteria have been fulfilled and the final reports of the results of the tests are compiled, the activities of the closure begin to occur. Table 7 shows the tasks included.

5 Conclusions

Automated tools and simulators do not necessarily produce reliable results since not all the accessibility problems can be automatically detected. Besides, a tool can produce fail positives and fail negatives, up to 33% and 35% respectively according to Brajnik (2). These fail positives and fail negatives need to be discarded by using a combination of tools (18) and including expert-based

evaluation and user-based testing. Hence, automated tools provide good support to software testers but they should be used in the context of a comprehensive method.

The method presented in this study is based on the five stages defined by ISTQB and adds specific activities related to accessibility testing of web applications in agile contexts.

Debuggers have to solve the accessibility issues found during accessibility testing by making changes on the web application to improve accessibility based on the evaluations results.

In the future, the definition of further requirements for each type of disability is necessary, as well as to propose mechanisms to overcome the challenges associated with the implementation, testing, and debugging of these requirements.

ID	TASKS	ACTIVITY
TCA1	Confirm test deliverables, final resolution or postponement of defect reports and acceptance of the system by receiving parties.	Closure
TCA2	Finalize and archive testware, test environment, and test infrastructure for later use during maintenance.	
TCA3	Deliver the testware and the possibility of additional items to the maintenance organization	
TCA4	Conduct a retrospective study to take into account improvements for future versions, projects and testing processes.	
TCA5	Store accessibility testing assets in the software configuration repository.	Accessibility

Table 7. Test closure activities.

References

(1) Brajnik, G. (2006). Web accessibility testing: when the method is the culprit. In *Proceedings of the International Conference on Computers for Helping People with Special Need (ICCHP),* pp. 156-163.

(2) Brajnik, G. (2008). Beyond conformance: the role of accessibility evaluation methods. In *Proceedings of the Web Information Systems Engineering Conference (WISE)*, pp. 63–80.

(3) Freire A. P., Goularte R., and Mattos-Fortes R. P. (2007). Techniques for developing more accessible web applications. In *Proceedings of the 25th ACM International Conference on Design of Communication (SIGDOC)*, pp. 162-167.

(4) Goncalves de Branco, R., Cagnin, M. I., Barroso Paiva, D. M. (2014). AccTrace: Accessibility in phases of requirements engineering, design, and coding software. In *Proceedings of the 14th International Conference on Computational Science and Its Applications (ICCSA)*, pp. 225-228.

(5) Hambling, B. (2015). Software Testing: An ISTQB-BCS Certified Tester Foundation Guide 3rd Edition. Swindon: BCS.

(6) Henka, A. and Zimmermann, G. (2014). Persona based accessibility testing. In *Proceedings of the International Conference on Human-Computer Interaction*, pp. 226-231.

(7) Herramhof, S., Petrie, H., Strobbe, C., Vlachogiannis, E., Weimann, K., Weber, G., & Velasco, C. A. (2006). Test case management tools for accessibility testing. In *Proceedings of the 10th International Conference on Computers for Helping People with Special Needs (ICCHP)*, pp. 215-222.

(8) ISO. (2011). ISO/IEC 25010 Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models. Retrieved from http://www.iso.org/iso/catalogue_detail.htm?csnumber=35733

(9) ISO. (2012). ISO 9241-171 Ergonomics of human-system interaction – Guidance on software accessibility. Retrieved from https://www.iso.org/obp/ui/#iso:std:iso:9241:-171:ed-1:v1:en

(10) Keates, S., Looms, P. O. (2014). The role of simulation in designing for universal access. In *Proceedings of 8th International Conference of Universal Access in Human-Computer Interaction (UAHCI)*, pp. 54-63.

(11) Luján-Mora, S. and Masri, F. (2012). Integration of Web Accessibility into Agile Methods. In *Proceedings of the 14th International Conference on Enterprise Information Systems (ICEIS)*, 3, pp. 123-127.

(12) Masri, F. and Luján-Mora, S. (2011). A Combined Agile Methodology for the Evaluation of Web Accessibility. In *Proceedings of the IADIS International Conference Interfaces and Human Computer Interaction 2011 (IHCI)*, pp. 423-428.

(13) Nvaccess. (2015). NVDA Screen Reader. Retrieved from http://www.nvaccess.org/

(14) Sánchez-Gordón, M-L., Moreno, L. (2014). Toward an integration of web accessibility into testing processes. In *Procedia Computer Science*, *27*, pp. 281-291.

(15) Sanchez-Gordon, S., Sánchez-Gordón, M-L., Luján-Mora, S. (2016).Towards an engineering process for developing accessible software in very small entities. In *Proceedings of the 11th International Conference on Evaluation of Novel Software Approaches to Software Engineering (ENASE)*, p.241-246.

(16) Slatin, J., Rush, S. (2003). Maximum accessibility: making your web site more usable for everyone. Addison-Wesley Professional.

(17) Torkey. F., Keshk, A., Hamza, T., and Ibrahim. A. (2007). A new methodology for web testing. In ¨*Proceedings of the 5th International Conference on Information and Communications Technology*, pp. 77–83.

(18) Vigo, M., Brown, J. and Conway, V. (2013). Benchmarking Web Accessibility Evaluation Tools: Measuring the Harm of Sole Reliance on Automated Tests. In *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility*, pp. 1-10.

(19) W3C. (2005). Introduction to web accessibility. Retrieved from http://www.w3.org/WAI/intro/accessibility.php

(20) W3C. (2008). Web content accessibility guidelines WCAG 2.0. Retrieved from http://www.w3.org/TR/WCAG20/

(21) W3C. (2010). Involving users in evaluating web accessibility. Retrieved from https://www.w3.org/WAI/eval/users.html

(22) W3C. (2013). Markup Validation Service. Retrieved from http://validator.w3.org.

(23) W3C. (2016). Web accessibility evaluation tools list. Retrieved from https://www.w3.org/WAI/ER/tools/

(24) WebAIM. (2015). WAVE Chrome Extension. Retrieved from http://wave.webaim.org/extension/

(25) WHO. (2011). World report of disability. Retrieved from http://www.who.int/disabilities/world_report/2011/report.pdf.

(26) Zimmermann, G., and Vanderheiden, G. (2008). Accessible design and testing in the application development process: considerations for an integrated approach. *Universal Access in the Information Society*, *7*(1-2), pp. 117-128.