Received: 20 May 2018 | Revised: 2 July 2018 | Accepted: 12 July 2018

DOI: 10.1002/smr.1987

SPECIAL ISSUE PAPER

WILEY Journal of Software: Evolution and Process

Integration of accessibility design patterns with the software implementation process of ISO/IEC 29110

Sandra Sanchez-Gordon[1] | Mary Sánchez-Gordón[2] | Murat Yilmaz[3] | Rory V. O'Connor[4]

[1] Escuela Politécnica Nacional, Quito, Ecuador

[2] Østfold University College, Halden, Norway

[3] Çankaya University, Ankara, Turkey

[4] Dublin City University, Dublin, Ireland

Correspondence
Mary Sánchez-Gordón, Østfold University College, Halden, Norway.
Email: mary.sanchez-gordon@hiof.no

Abstract

The Web Content Accessibility Guidelines was developed by World Wide Web Consortium with a goal of providing a single shared standard for web content accessibility that meets the needs of individuals, organizations, and governments. Given that there is a large percentage of very small entities that develop software who also utilize the ISO/IEC software process standard, the purpose of this study is the development of software design patterns for users with visual disabilities. As a result, four accessibility design patterns are defined: Authentication adapter, Blindness adapter, Dichromatic color vision adapter, and Blurry vision adapter. These patterns will help to improve the design of the web applications built using them while being compliant with the ISO/IEC 29110 standard. The use of design patterns also enables the transfer of design experience to programming practices and improves the software documentation. To validate the set of patterns, an online course for Spanish speakers was developed, and the evaluation was carried out using simulators, automated tools, experts, and users. Simulators and automated tools showed no accessibility errors and experts evaluated 10 heuristics principles and did not identify any severity issues. Taken together, our results provide positive evidence that users with visual disabilities could benefit from the proposed features.

KEYWORDS

ISO/IEC 29110, software design patterns, software process, very small entities, web accessibility, web content accessibility guidelines

1 | INTRODUCTION

The desired characteristics of a software product under development are strongly influenced by situational factors that affect the software development process being used.[1] One of these important characteristics is software quality which needs to be taken in account to tailor the activities and outputs of the software development process.[2] In particular, Accessibility is a quality attribute defined as a software quality sub-factor of Usability according to the standard ISO/IEC 25000:2014 "Systems and Software Engineering—Systems and Software Quality Requirements and Evaluation."[3] Thus, accessibility is the usability of a product, service, environment, or facility by people with the widest range of capabilities.[4] In this widest range, people with visual disabilities constitute an important user group to consider because of the 7.33 billion people alive in 2015, an estimated 36 million were blind, 217 million were moderately or severely vision impaired, and 188 million had mild vision impairment.[5] The positive effects of accessibility on nondisabled users were studied by Schmutz et al[6] the findings revealed that Web Content Accessibility Guidelines (WCAG) can support users with and without visual disabilities alike. Therefore, a wide range of users can positively benefit by accessible software. Particularly, according to Sanchez-Gordon et al[7] a design for a wide range of users provides advantages that may change the perception of practitioners in a positive way, moving from an "accessibility for users with disabilities" approach to "inclusive-design." Moreover, implementing accessibility is part of the legal requirements in several countries.[6,7] However, in spite of legal regulations on accessibility

and detailed guidelines that exist since many years, the application of WCAG, which is the most widely used standard among researchers and practitioners, in practice is still rare.[6,8] According to Schmutz et al,[8] in recent studies, more than 95% of websites investigated were classified as not accessible, which means that the specific needs of people with different forms of visual disabilities, including blindness, are not sufficiently considered.[9]

Despite the fact that there are a large percentage of software companies classified as very small entities (VSEs), relatively few studies have focused on the actual processes that they are using.[10] A VSE is defined as "an enterprise, an organization, a department or a project having up to 25 people"[11] and according to Sanchez-Gordon et al,[7] VSEs account for 85% of software development organizations in the European Union, 57% in the United States, 80% in Canada, and 90% in Ecuador. Moreover, it has also been recognized that most VSEs cannot afford the resources—in number of employees, expertise, cost, and time—or see a net benefit in establishing software processes.[10] However, the need for all organizations—not just VSEs—to pay attention to software process practices such as ISO standards have been established.[12] In fact, most of the related research efforts about the ISO/IEC 29110 standard had focused in software process improvement (SPI), particularly project management (PM). Mesquida and Mas[13] describe the lessons learned from the implementation of the ISO/IEC 29110 standard integrated with the PMBOK Guide in four software development VSEs. The main result is the development of a process asset library to support PM best practices. Larrucea et al[14] explain that large software development organizations typically use traditional SPI models, such as CMMI, but VSEs cannot afford such implementations. Instead, this study proposes building an experience factory that will help VSEs start SPI initiatives to improve software quality. For VSEs, the quality of their software products is a key factor for its competitive advantage.[15] However, the results from the study performed by Larrucea et al[16] reveal that one of the areas where organizations that uses ISO/IEC29110 have problems most of the time are related to software quality assurance, along with change request, software configuration, and verification and validation activities.

In this scenario, the issue is that once a VSE adopts the ISO/IEC 29110 standard, it does not have support from the software implementation (SI) process to implement accessibility requirements in the software products being developed. Indeed, there are very few studies focused on accessibility in software engineering. For example, De Branco et al[17] state that providing accessible web software is a complex challenge, especially the implementation of requirements since many developers might not have the necessary skills to perform this task. As a solution, a tool for managing traceability of accessibility requirements from conception to the coding phases is presented. In the specific context of integrating ISO/IEC 29110 standard and accessibility, a previous study[7] conducted an evaluation of six software products developed by VSEs and the negatives results revealed that there is a need in VSEs of an engineering process which takes in account their particular features and can be applied for them to assure the accessibility of their software. In light of that, the aim of this study is to present an approach to integrate accessibility tasks into the ISO/IEC 29110 standard that will enable the integration of a set of proposed accessibility patterns within the standard.

The remainder of this paper is organized as follows. Section 2 introduces insights and related literature on ISO/IEC 29110, design patterns, disability, accessibility, and WCAG. Section 3 presents the four design patterns for accessibility proposed in this study. In Section 4, the suggested tasks for the implementation of accessible web application using ISO/IEC29110 standard and the proposed software patterns are presented. In Section 5, a case study about accessible online English course for Spanish speakers with visual disabilities is described, including design and results. Section 6 presents a discussion and threats of validity. Finally, Section 7 offers main conclusions and future research directions.

2 | THEORETICAL BACKGROUND

2.1 | ISO/IEC 29110

The origins of the ISO/IEC 29110 standard go back to 2004 when the ISO sub-committee SC7 mandated to develop software engineering standards suited for VSEs.[18] There is a general recognition of the specific adoption issues that VSEs face due to their sizes, business models, situational factors, and risk levels.[19,20] Part 1 of the ISO/IEC 29110 standard provides an overview of the standard introducing processes, lifecycle and standardization concepts, and the characteristics and requirements of a VSE.[21] The core of the standard is in Part 5, which is a Management and Engineering guide focusing on two main processes: project management (PM) and software implementation (SI). An interesting approach proposed in this standard to assist VSEs in their deployment is the development of a series of deployment packages (DP). A DP includes a set of artifacts, such as a description of processes, activities, tasks, steps, roles, products, templates, checklists, examples, references and mapping to standards and models, and a list of tools.[22] These DPs are defined as four ISO/IEC 29110 profiles included within a generic profile group: Entry, Basic, Intermediate, and Advanced, which are generic profiles applicable to VSEs that do not develop critical software. They were published between 2011 and 2018.[22-25] VSEs can achieve international recognition for the quality of their software products through implementing one of the generic profiles and by being audited against ISO/IEC 29110 specifications using the assessment guide included in the standard.[26,27]

The present study is focused on the integration of accessibility design patterns within the SI process of the ISO/IEC 29110. The purpose of SI is to achieve systematic performance of the analysis, design, construction, integration, and test activities for software products according to a set of specified requirements, including quality requirements such as accessibility. The SI process can be used with any lifecycle such as waterfall, iterative, incremental, evolutionary or agile. Figure 1 illustrates the six activities of the SI process: (1) implementation initiation, (2) requirements analysis, (3) architectural and detailed design, (4) construction, (5) integration and tests, and (6) product delivery, including their relevant work products and relationships.[22]

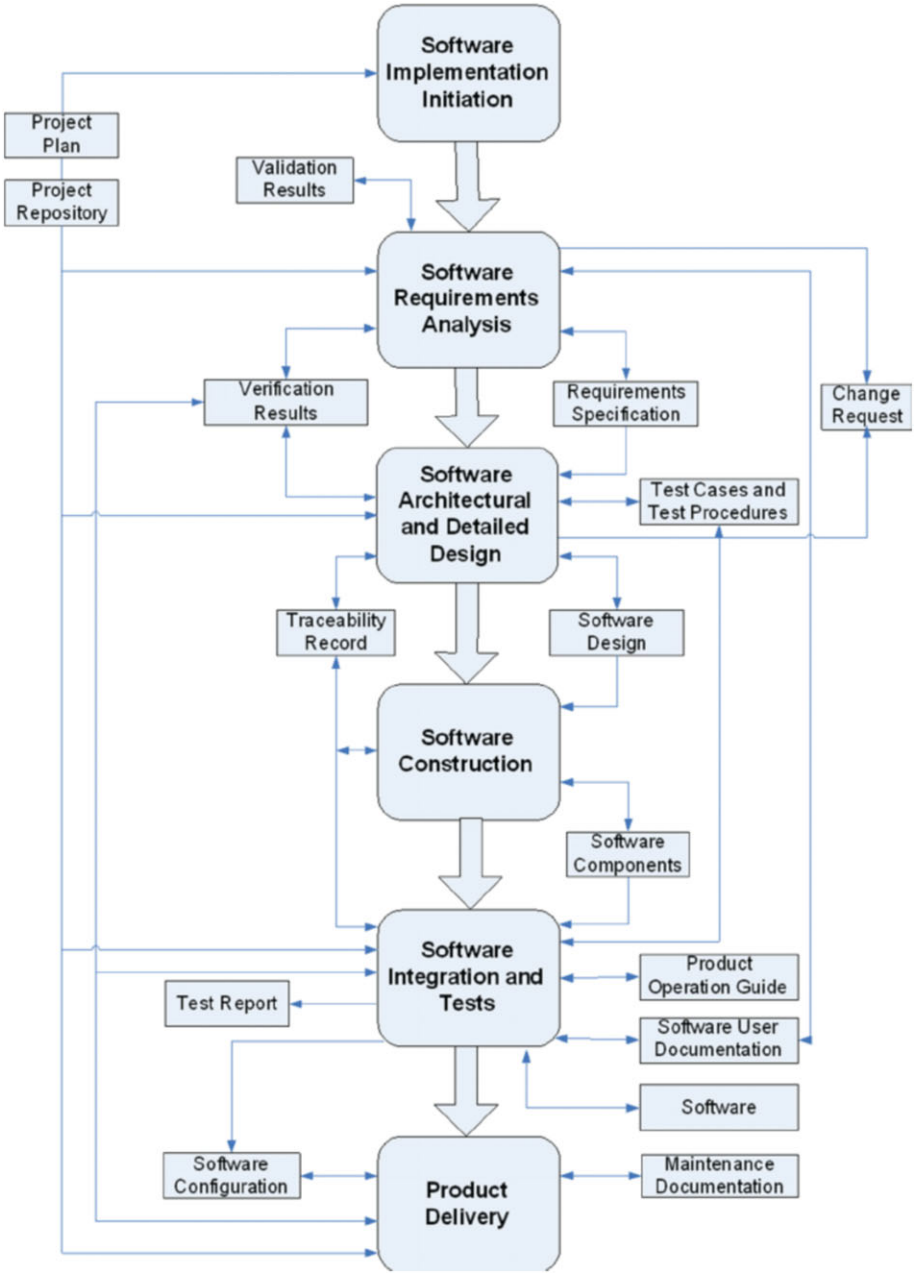

FIGURE 1 ISO/IEC 29110 software implementation process[22]

2.2 | Design patterns

A design pattern arises due to the appearance of a recurrent problem in a given environment, eg, the web environment.[28] A formal definition states: "A design pattern names, abstracts and identifies key aspects of a common design structure that make it useful to create a reusable object-oriented design. The design pattern identifies the classes and instances involved their roles and collaborations, and the distribution of responsibilities. Each design pattern focuses on a problem, usually an object-oriented design problem. It describes when it is applied, if it can be applied in view of other design limitations, and the consequences and commitments to use."[29]

Design patterns enable software designers to manage design knowledge for later reuse. In 1964, the architect and mathematician, Christopher Alexander, was the first to write about patterns, and in 1987, Ward Cunningham and Kent Beck adopted this approach to object-oriented programming and user interfaces implementation. Design patterns made their breakthrough in software engineering in 1995 when Gamma et al published their classic book "Design Patterns: Elements of Reusable Object-Oriented Software."[29] Since then, many authors have proposed software design patterns using different structures. The structure for a design pattern must include certain parameters to make it useful. The same software design pattern can be represented in different ways to visualize more extensive or more simplified structures. In the context of this study, a specific structure for the four accessibility patterns has been defined with a set of basic elements: name, context of use, problem, forces, solution, implementation, pros, cons, quality factors, and metrics. This set of elements has been based on the proposals in the literatures.[30-35]

2.3 | Disability, accessibility, and WCAG

The United Nations' Convention on the Rights of Persons with Disabilities recognized that "disability is an evolving concept and that disability results from the interaction between persons with impairments and attitudinal and environmental barriers that hinders their full and effective participation in society on an equal basis with others."[36] Disability is not an attribute of the person, but it depends on the barriers that persons with disabilities encounter in their day-to-day lives. In 2011, the WHO/World Bank's World Disability Report found that the global prevalence of disability was 16%, ranging from 12% in higher income countries to 18% in lower income countries.[37]

In this study, three types of visual disabilities were considered: blindness, dichromatic color vision, and blurred vision. Blindness is a term used when for people without usable vision, or only with the ability to perceive light. People with this visual disability may be born with no vision, or they develop vision loss later in life as a result of an accident or illness. In 2015, an estimated 253 million people live with vision disabilities in the world: 36 million are blind and 217 million have moderate to severe vision disability.[5] Dichromatic color vision is a deficiency in seeing color, with varying causes including a genetic condition, a consequence of diseases such as diabetes or multiple sclerosis, or be due to the aging process. It affects approximately one in 12 men (8%) and one in 200 women in the world (0.5%). People with dichromatic color vision have difficulty in seeing the difference between red and green or between blue and yellow.[38] Blurred vision is the loss of sharpness of sight, making objects appear out of focus and confusing. The primary causes of blurred vision are refractive errors, myopia, hyperopia, astigmatism, or presbyopia. Blurred vision can also be a symptom of more serious problems, such as eye diseases or neurological disorders.[39]

Accessibility refers to the provision of flexible facilities and environments, either virtual or physical, to accommodate every user's needs and preferences. An accessible design ensures unassisted "direct access" or "indirect access" through assistive technology, such as screen readers.[36] The US Department of Education Office for Civil Rights states: "Accessible means a person with a disability is afforded the opportunity to acquire the same information, engage in the same interactions, and enjoy the same services as a person without a disability in an equally effective and equally integrated manner, with substantially equivalent ease of use. A person with a disability must be able to obtain information as fully, equally and independently as a person without a disability".[40] Therefore, accessibility seeks to generate equity and a more inclusive society, and in the case of web accessibility, it also increases usability for all users.[41]

The Web Accessibility Initiative of the World Wide Web Consortium created the WCAG, with the latest version, WCAG 2.1, being released as a recommendation in 2018. The WCAG covers a range of barriers for accessibility of web content organized in four principles, 12 guidelines, and three levels of conformance. The principles that provide the foundation for web accessibility are as follow: perceivable, operable, understandable, and robust. Perceptible means that the information and the components of the user interface must be available for different human senses. Operable means that users must be able to interact with the different controls and interactive elements using a mouse, keyboard, or other devices. Understandable means that the web content must be clear, written without unusual words, and the web pages must operate in a predictable manner. Robust means that the web content must be easy to access using a wide range of technology, including current and future applications and assistive technology. For each principle, there is a set of guidelines. Each guideline has a number of testable success criteria and techniques to achieve them, which fall into two categories: those that are sufficient for meeting the success criteria and those that are advisory. Finally, the levels of conformance are: A (lowest), AA (intermediate), and AAA (highest). In this study, the levels applied are A and AA.[42]

3 | DESIGN PATTERNS FOR ACCESSIBILITY

The four accessibility design patterns defined in this study are (1) authentication adapter, (2) blindness adapter, (3) dichromatic color vision adapter and (4) blurry vision adapter. These patterns are based on the adapter pattern. The adapter pattern is a classic software design pattern where an adapter makes one interface (the adaptee's) conforms to another, thereby providing a uniform abstraction of different interfaces. A class adapter accomplishes this by inheriting privately from an adaptee class. The adapter then expresses its interface in terms of the adaptee's.[28] The accessibility patterns adapt the interface and content of web applications according to WCAG, providing a good user experience for blindness, dichromatic color vision, and blurry vision users. The Authentication Adapter allows the login and registration of users. By default, the authentication interface of a web application must be accessible to all types of users since the user profile will only be available once the user is registered. The Blindness Adapter allows the information displayed through a web application to be accessible, regardless of the browser, screen reader, and device in which it is displayed. The Dichromatic Color Vision Adapter transforms color-based information for users who do not have a proper distinction between red, green, blue, and yellow. The Blurry Vision Adapter implements the guidelines for displaying information in a way appropriate for low vision users.

To develop the patterns, a mapping was performed between the accessibility requirements for each of the three visual disabilities and the recommendations to solve them as proposed by the WCAG guidelines. Table 1 presents the mapping for blind users as an illustration, while the other two mappings are not included due to space constraints.

Once the mapping was completed, the next step was to document the patterns using the structure previously defined. Table 2 shows the complete documentation for the accessibility adapter for blindness users. The documentation for the three other patterns is not included due to space constraints.

The UML class diagrams for each accessibility pattern, including attributes and methods, were developed. Figure 2 presents, as an example, the UML class diagram for the blindness adapter.

TABLE 1 Requirements and accessibility guidelines for blind users

Accessibility Requirements for Blind Users	Pertinent WCAG Guidelines and Levels
● All functionality must be operable by the keyboard. ● Images, sensitive areas of image maps and other non-text elements must have self-explanatory alternative text since screen readers cannot perceive images. ● Videos must have textual narratives such as audio descriptions and extended captions. ● Animations and other moving content such as carrousels should not be used. ● Links must have significant titles and be unique within the same web page. ● Input forms must have labels for all the fields. ● Tables should be understandable when read sequentially. ● Tables should not be nested. ● All web pages must have a linear order and a coherent structure. ● Web pages must have headers of different levels. ● Web pages should not have programmatic events dependent on a click or mouse movement.	1.1.1 Text alternatives for non-text content (A) 1.2.1 Time-based media for audio-only and video-only (pre-recorded) (A) 1.2.3 Audio description or media alternative (prerecorded) (A) 1.2.5 Audio description (prerecorded) (AA) 1.3.1 Adaptable—Info and relationships (A) 1.3.2 Adaptable—Meaningful sequence (A) 1.4.2 Distinguishable—Audio control (A) 2.1.1 Keyboard accessible (A) 2.1.2 No keyboard trap (A) 2.2.1 Timing adjustable (A) 2.2.2 Pause, stop, hide (A) 2.4.1 Bypass blocks (A) 2.4.2 Page titled (A) 2.4.3 Focus order (A) 2.4.4 Link purpose (in context) (A) 2.4.5 Navigable in multiple ways (AA) 2.4.6 Headings and labels (AA) 3.1.1 Language of pages (A) 3.1.2 Language of parts (AA) 3.2.1 Predictable on focus (A) 3.2.2 Predictable on input (A) 3.2.3 Consistent navigation (AA) 3.2.4 Consistent identification (AA) 3.3.1 Error identification (A) 3.3.2 Labels or instructions (A) 3.3.3 Error suggestion (AA) 3.3.4 Error prevention (AA) 4.1.1 Parsing (A) 4.1.2 Name, role, value (A)

4 | INTEGRATION OF ISO/IEC 29110 SOFTWARE IMPLEMENTATION PROCESS AND ACCESSIBILITY PATTERNS

The goal of the integration of accessibility patterns with the SI process of the standard ISO/IEC 29110 is to develop a software product that satisfies the needs and expectations of all potential users, including users with accessibility needs due to permanent, temporal, or environmental disabilities, eg, users who are born blind, users with temporary blindness or blurred vision due to surgery, or users working under extreme light conditions.[42] The following subsections details the tasks associated to each activity of the SI process plus the additional tasks proposed to integrate the accessibility patterns.

4.1 | Software implementation initiation

The Software Implementation Initiation activity of ISO/IEC 29110 ensures that the Project Plan established in Project Planning activity is committed to by the Work Team. The tasks of this activity are[3]:

- The project manager and the work team review the project plan to achieve a common understanding and commitment.
- The work team sets or updates the implementation environment.

The additional tasks proposed to integrate the accessibility patterns are:

- The work team selects simulation aids for testing purposes; eg, blindfolds.
- The work team selects assistive technologies, eg, the ChromeVox screen reader.
- The work team selects simulators for different types of disabilities; eg, SimDaltonism simulates eight types of dichromatic color vision.[43]
- The work team selects accessibility evaluation tools, eg, AChecker, WAVE.[44]
- The work team selects HTML and CSS checkers, eg, World Wide Web Consortium HTML Validator.[45]

4.2 | Software analysis

The Software Requirements Analysis activity analyses the agreed customer's requirements and establishes the validated project requirements. The tasks of this activity are[3]:

TABLE 2 Blindness adapter structure

Element	Sub-Element	Request
Identification	Name	Accessibility adapter for blindness users
	Key	Blindness adapter
	Category	Structural
	Keywords	Blindness, accessibility
	Related patterns	Adapter Decorator Façade
Context of use	Users	Users with blindness, ie, users who cannot perceive any kind of visual information.
	Task	● Receive the specific request. ● Provide the adapted page for blindness users: ○ Facilitate keyboard navigation. ○ Define a coherent sequence in the content programming. ○ Implement alternative media for images and videos.
	Platform capabilities	The information displayed through a web page that implements the accessibility pattern for blindness users must be accessible regardless of the browser, screen reader, and device in which it is displayed.
Problem	Most web applications do not have a content accessibility level of type AA. Is it possible for blindness users to have the same user experience as users who don't have this disability?	
Forces	According to the WCAG guidelines[40]: ● Success criterion 1.1.1 Non-text content. ○ Add titles or descriptions to content components. ● Success criterion 1.2.1 Audio-only and video-only (prerecorded). ○ Implement time-based media alternatives (subtitles or audio descriptions). ● Success criterion 1.2.3 Audio description or media alternative (prerecorded). ○ Implement time-based media alternatives (subtitles or audio descriptions). ● Success criterion 1.2.5 Audio description (prerecorded). ○ Add audio descriptions to the content. ● Success criterion 1.3.2 Meaningful sequence. ○ Follow a consistent sequence in content programming. ● Success criterion 1.4.2 Audio control. ○ Add a mechanism to pause, stop, or control the volume of any audio that plays automatically in the web page for more than 3 s. ● Success criterion 2.1.1 Keyboard. ○ Implement attributes or components that facilitate navigation (*tabindex*). ● Success criterion 2.1.2 No keyboard trap. ○ Use the keyboard shortcuts recommended by the OS for greater accessibility. ● Success criterion 2.2.1 Timing adjustable. ○ The user can control every time limit that is set by the content. ● Success criterion 2.2.2 Pause, stop, hide. ○ Control information actions. ● Success criterion 2.4.1 Bypass blocks. ○ Bypass blocks of content that are repeated. ● Success criterion 2.4.2 Page titled. ○ Web pages have titles that describe topic or purpose. ● Success criterion 2.4.3 Focus order. ○ Implement a logical order component (*tabindex*). ● Success criterion 2.4.4 Link purpose (in context). ○ Implement a descriptive text for the link. ● Success criterion 2.4.5 Multiple ways. ○ Implement a navigation component (*href*).	

(Continues)

TABLE 2 (Continued)

Element	Sub-Element	Request
		● Success criterion 2.4.6 Headings and labels. ○ Headings and labels describe topic or purpose. ● Success criterion 3.1.1 Language of page. ○ Implement a language attribute (*lang*) ● Success criterion 3.1.2 Language of parts. ○ Implement a language attribute (*xml: lang*) ● Success criterion 3.2.1 On focus. ○ Implement a navigation component. ● Success criterion 3.2.2 On input. ○ Implement a navigation attribute (*submit*). ● Success criterion 3.2.3 Consistent navigation. ○ Implement a constant navigation component on all screens (title, footer). ● Success criterion 3.2.4 Consistent identification. ○ Implement an information component (*alt*). ● Success criterion 3.3.1 Error identification. ○ Implement a navigation attribute (*aria-invalid*). ● Success criterion 3.3.2 Labels or instructions. ○ Implement a descriptive component (*aria-label*). ● Success criterion 3.3.3 Error suggestion. ○ Implement a requirement component (*aria-required*). ● Success criterion 3.3.4 Error prevention (legal, financial, data). ○ Implement a confirmation component (*checkbox*). ● Success criterion 4.1.1 Parsing. ○ Implement a validation component (*xmlvalidate*). ● Success criterion 4.1.2 Name, role, value. ○ Implement a navigation component (*aria-label*).
Solution		Adapt the content of the web pages according to the accessibility guidelines for universal web design in order to provide a good experience for blind users regarding the user-application interaction.
Implementation	Structure	See UML class diagram in Figure 2. Available online: http://creately.com/diagram/example/j4tbuwf52
Consequences	Pros	● Blindness' accessible web pages. ● Specific adaptations for blind users. ● Increment web application usability.
	Cons	● Longer development time. ● Updates to the pattern will be necessary if the WCAG guidelines changes.
	Quality factors and metrics	● Functional suitability measures The pattern conforms to the accessibility requirements of blind users. The list below presents the suitability measures that can be applied to the blindness pattern. ○ Functional completeness measures. ○ Functional correctness measures. ○ Functional appropriateness measures. ● Usability measures The pattern improves the user experience of blind users. The list below presents the usability measures that can be applied to the blindness pattern: ○ Operability measures. ○ Accessibility measures. ○ Learnability measures.

(Continues)

TABLE 2 (Continued)

Element	Sub-Element	Request
		● Any web application can implement the pattern without difficulty. It can be evaluated by: ○ Maturity measures. ○ Availability measures. ○ Maintainability measures. ● Maintainability measures The pattern improves the maintainability of the web application. The list below presents the maintainability measures that can be applied to the blindness pattern: ○ Modularity measures. ○ Reusability measures. ○ Analyzability measures. ○ Modifiability measures. ○ Testability measures. ● The blindness pattern is an architectural guide and can be adapted by any developer according to the particular needs and circumstances.

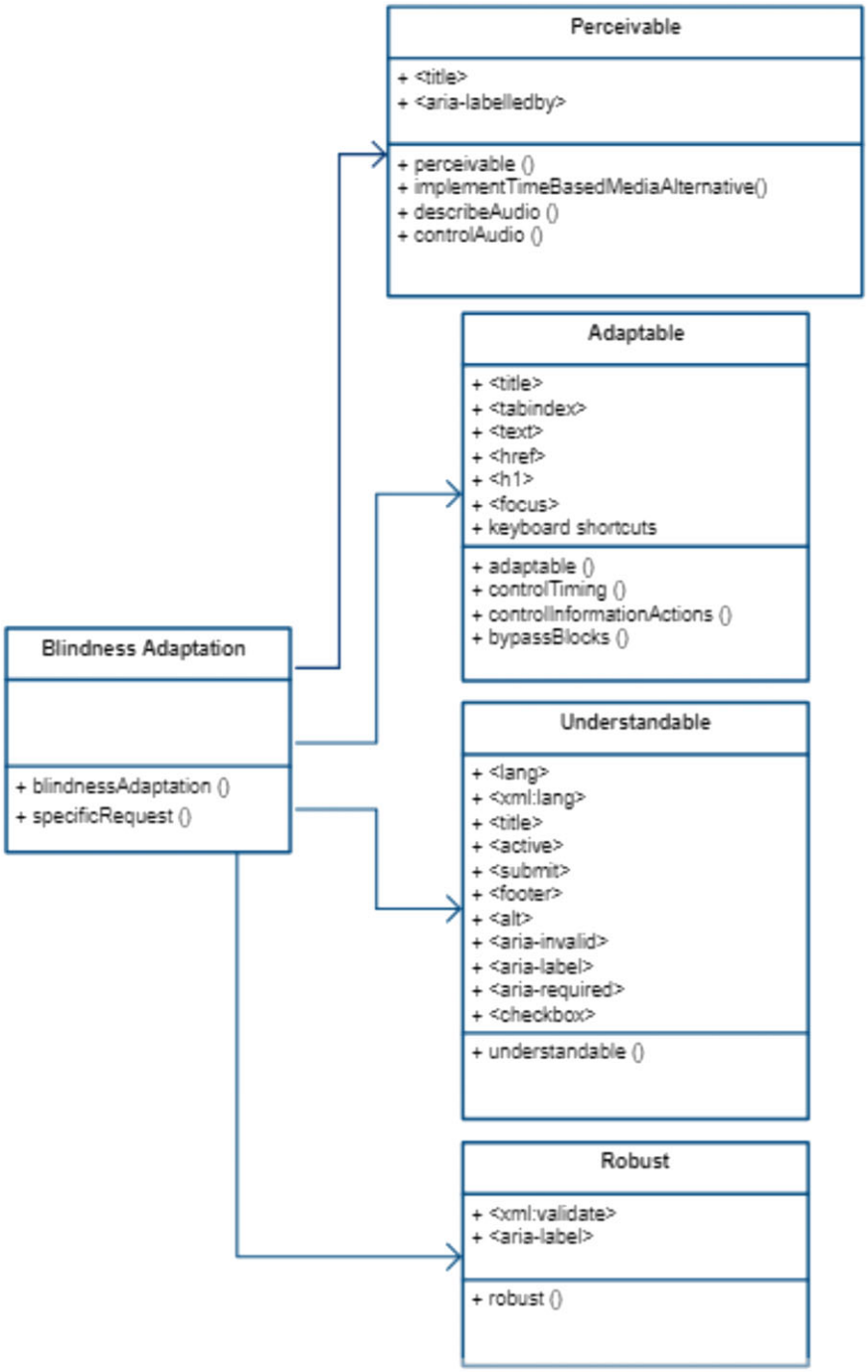

FIGURE 2 Blindness adapter UML class diagram

- The project manager and work team assign tasks to each team member according to their role and the project plan.
- The work team identifies and consults information sources.
- The work team and customer determine the scope and feasibility, verify the correctness and testability, and generate/update the Requirements Specification.
- The customer validates and obtains approval of the Requirements Specification and validates the Requirements Specification traceability with the needs and expectations.

The additional tasks proposed to integrate the accessibility patterns are:

- The project manager sensitizes the work team through the observation of users with disabilities interacting with software products.
- The work team uses cost-effective approaches to gather and validate accessibility requirements based on WCAG guidelines.
- The work team and customer include accessibility requirements in the Requirements Specification.

4.3 | Software Component Identification

The tasks of this activity are[3]:

- The project manager and work team assign tasks to each team member according to their role and the project plan.
- The work team understands the Requirements Specification.
- The work team creates/updates the Software Component Identification, analyses the Requirements Specification to generate the components, and provides details of the software components and their interfaces to allow the construction.

The additional tasks proposed to integrate the accessibility patterns are:

- The work team defines and reviews the design architecture, software components, and interfaces for traceability with accessibility requirements in the Requirements Specification.
- The work team includes the accessibility patterns in the design architecture, software components, and interfaces.
- The project manager guides the team members in getting accessibility knowledge and using techniques for building accessible software components, eg, the use of Web Accessibility Initiative ARIA in HTML5.[46]

4.4 | Software Construction

The Software Construction activity develops the software code and data from the Software Design. The tasks of this activity are[3]:

- The project manager and work team assign tasks to the team according to their role and the plan.
- The work team understands the Software Component Identification.
- The work team constructs/updates the Software Components.
- The work team creates/updates the Test Cases and Test Procedures for unit and integration. The customer provides testing data.
- The work team tests the Software Components. Correct the defects found until successful unit test is achieved.

The additional tasks proposed to integrate the accessibility patterns are:

- The work team builds accessibility features in the Software Components according to the accessibility patterns.
- The work team creates Test Cases and Test Procedures for testing the accessibility requirements.
- The work team uses accessibility checklists, eg, WebAIM WCAG Checklist.[47]

4.5 | Software Integration and Tests

The tasks of this activity are[3]:

- The project manager and work team assign tasks to each team member according to their role and the plan.
- The work team understands the Test Cases and Test Procedures.
- The work team sets/updates the testing environment.
- The work team integrates the software using Software Components and updates Test Cases and Test Procedures for integration testing.
- The work team performs software tests using Test Cases and Test Procedures for integration and creates the Test Report.
- The work team corrects the defects found until successful test is achieved.
- The work team incorporates the Requirements Specification and software to the Software Configuration.

The additional task proposed to integrate the accessibility patterns are:

- The work team uses the selected accessibility evaluation tools in the testing environment.

4.6 | Product delivery

The Product Delivery activity provides the integrated software product to the Customer. The tasks of this activity are[3]:

- The project manager and work team assign tasks to each team member according to their role and the plan.
- The work team reviews the Software Configuration for understandability.
- The project manager and work team perform delivery to the project manager and support delivery according to the project plan.

The additional task proposed to integrate the accessibility patterns are:

- The work team includes the accessibility assets in the Software Configuration.

5 | CASE STUDY: ACCESSIBLE ONLINE ENGLISH COURSE FOR SPANISH SPEAKERS WITH VISUAL DISABILITIES

5.1 | Scope

For this case study, the web application developed is an online course for learning English designed for Spanish speakers with the three visual disabilities considered in this study. This web application was developed using the proposed implementation process integrated with accessible patterns. Moreover, although the needs of blind people are well known and reflected in existing recommendations,[42] the requirements specification was defined with the participation of two blind users.

5.2 | Construction

The accessible online course implements the registration of users including their disability, as shown in Figure 3.

This registration enables the sending of a request to display the online course interfaces with the accessibility adaptions needed by the authenticated user using a CSS style sheet. Figure 4 shows a snippet of the PHP code that performs the request.

As an example, Figure 5 displays the main interface of the online course, with and without the on-the-fly adaptation for a blurred vision user. The adaptation changes the font type and size and background/foreground color contrast.

The online course includes registration, authentication, course lessons, evaluation, and evaluation feedback for each lesson. The web application is hosted in http://codelearncode.com/prototype/home.php.

5.3 | Tests

The evaluation of the web application was carried out with (1) visual disabilities simulators and accessibility evaluation automated tools, (2) heuristic evaluation with experts, and (3) usability testing users.

We executed the tests on personal computers running the Windows 8 operating system, equipped with a 17-in monitor, a common keyboard with the letter part on the left side and the block of numbers on the right side, a two-button mouse, and a set of headphones. Blind users did not need the monitor or the mouse since they used exclusively keyboard navigation and the headphones to listen to the screen reader. One blind user used the ChromeVox screen reader version 53.0.2784.5 on and Google Chrome browser version 67.0.3396.99, and the second blind user used the NVDA screen reader version 2015.2 on Internet Explorer 11 browser.

The three visual simulators used were NoCoffee, Vision and Hearing Impairment, Simulator and Spectrum, and the two accessibility evaluation tools used were WAVE and AChecker. NoCoffee is an extension for Chrome navigator that filters the displayed web pages according to the configuration of pervasive visual issues such as low acuity (blur), contrast loss, glare, and different types of color blindness. The tests performed with

FIGURE 3 Web interface for user registration

```
<?php
if ((adapter($userRow['userDis'])=='blindness')){
$disability=adapter($userRow['userDis']);
}else if ((adapter($userRow['userDis'])=='dichromatic')){
$disability=adapter($userRow['userDis']);
}else if ((adapter($userRow['userDis'])=='blurry')){
$disability=adapter($userRow['userDis']);
}else {
$disability=adapter($userRow['userDis']);
}
?>
<link rel="stylesheet" href="Patterns/<?php echo $disability?>.css"
type="text/css" />
```

FIGURE 4 Snippet of the adaptation programmed logic

FIGURE 5 Main interface on the online course with and without adaptation for blurred vision users

NoCoffee showed that the web application is accessible for users with blurry vision up to grade 3. Similarly, NoCoffee showed that the web application is accessible for users with the eight types of dichromatic color vision that the tool simulates, ie, protanopia, protanomaly, deuteranopia, deuteranomaly, tritanopia, tritanomaly, achromatopsya, and achromatomaly. The Vision and Hearing Impairment Simulator is a tool that applies simulated vision impairment to images or software. Tests performed with this simulator showed that the online course is accessible for users with blurred vision due to cataracts of 80% severity. Spectrum is an extension for Chrome navigator to test web pages for users with different types of color vision deficiencies. The test performed showed the same results as the test performed using NoCoffee. WAVE is an extension of Chrome navigator that presents the evaluated web page with embedded icons and indicators that visually present information about the accessibility. The results obtained were zero errors and zero alerts. AChecker is an open tool used to evaluate web pages for accessibility issues and identifies three types of problems: know, likely, and potential. The results of the tests executed indicated zero known problems, zero likely problems, and 31 potential problems. As an example, Figure 6 shows the results of the accessibility evaluation with WAVE and the simulation with NoCoffe.

Three experts collaborated with the heuristic evaluation phase. Their average of years of experience in software development is 13.6, and their average of years of experience in usability and accessibility evaluations is 5.6. We used an evaluation template based on Nielsen's 10 heuristics principles[48] and a three-point scale: (1) not fulfilled, (2) partially fulfilled, and (3) fully fulfilled. This scale helped not only to assess whether the interfaces comply with a certain heuristic or not, but also to identify their degree of severity. The degree of severity was determined according to the following criteria:

- High Severity: When there is coincidence of criteria of the three experts.
- Medium Severity: When there is coincidence of criteria of two experts.
- Low Severity: When there is no majority.

FIGURE 6 Accessibility evaluation results with WAVE and simulation with NoCoffee

Table 3 summarizes the results of the heuristic evaluation with experts. There were no high severity issues. The two medium severity issues pointed out to partially fulfilled heuristics related to instructions and documentation to use the web application.

For the user usability study, we used a user sample consisting of 14 participants including two blind people. Table 4 details the ethnography information of the participants' sample. This sample is considered sufficient to establish a preliminary tendency in the user experience, as stated by the 10 ± 2 rule of Hwang and Salvendy explained in Schmettow.[49]

The usability study was carried out in a user experience laboratory setting at *Escuela Politécnica Nacional* based on the strategy explained by Rubin and Chisnell.[50] We used a performance measurement testing technique and an interview as the inquiry method for the users.[51] We defined the users' objectives, steps to complete tasks, and questions to evaluate in the interview. Each session was conducted with one participant at a time. We explained to the participants that they were playing the role of testers rather than being the tested subjects. We designed a template with a list of task to be performed by the user shown in Table 5.

In addition, we defined four simple questions to collect information about the participants experience and to evaluate the completion of the list of tasks, as shown in Table 6. Only in one case did a user identify a difficulty in the action being executed.

6 | DISCUSSION

The findings of this study revealed that the proposed software process allows the development of a web application according to ISO/IEC 29110 standard and compliant with the AA accessibility level of WCAG. This proposed software process is potentially significant because there is a large percentage of VSEs that develop software around the world and the WCAG is also the basis of legal requirements for web accessibility in several regions and countries, such as Argentina, Australia, Brazil, Canada, China, Colombia, Ecuador, France, German, India, Ireland, Italy, Japan, Norway, New Zealand, Spain, United States, and United Kingdom. The findings are in line with previous work that indicated positive effects of complying with WCAG to improve the level of accessibility.[6,8,52] Those previous studies found also that both nondisabled and disabled users would benefit from higher accessibility levels, so this is also expected (although not explored here) for our web application. Moreover, it is worth noting that principles 2 and 6 of SPI Manifesto[53] can be distinguished in the proposed software process. The SPI Manifesto second principle states "motivate all people involved" was present when we provided the necessary resources such as training, equipment, and coaching to all people who had to use the proposed process for the development of the web application including the two blind users that participated in the specifications of

TABLE 3 Results of heuristic evaluation

Heuristic	Definition	Not Fulfilled	Partially Fulfilled	Fully Fulfilled
Visibility of system status	The system should always keep users informed about what is going on, through appropriate feedback within a reasonable time.	0	0	3
Match between system and the real word	The system should speak the users' language, with words, phrases, and concepts familiar to the users. Make the information appear in a natural and logical order.	0	0	3
User control freedom	"Emergency exit" to leave the unwanted state without having to go through an extended dialog. Support undo and redo.	0	1	2
Consistency and standards	Users should not have to wonder whether different words, situations, or actions mean the same thing.	0	0	3
Error prevention	Even better than good error messages is a careful design which prevents a problem from occurring in the first place.	0	0	3
Recognition rather than call	Instructions for use of the system should be visible or easily retrievable whenever appropriate.	1	2	0
Flexibility and efficiency of use	Allow users to tailor and speed up frequent actions.	0	0	3
Esthetic and minimalist design	Dialogs should not contain information which is irrelevant or rarely needed.	0	0	3
Help users recognize, diagnose, and recover from error	Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.	0	0	3
Help and documentation	Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.	1	2	0

TABLE 4 Participants' ethnography information

Participant	Background	Gender	Age	Computer Literacy
Blind user 1	School teacher. At 14 years of age, Franklin received a blow to the head which made him lose his sight.	Male	37	Medium user with 8 years of experience.
Blind user 2	Harvard alumni and inclusive education advocate. Nicholas lost his vision as a child.	Male	35	Advanced user with 20 years of experience.
Users with simulators and blindfolds	College students	8 male 4 female	24.5	Advanced users with 15.5 years on average.

TABLE 5 List of tasks for the usability study

#	Task	Solve without Help (S) Solved with Help (H) Unsolved (U)	Time Employed (min)
1	Enter the URL of the online course.		
2	Go to the registration page.		
3	Input registration data and successful register as a new user.		
4	Authenticate with login and password.		
5	Go to lesson 1 and navigate the educational contents.		
6	Go to evaluation for lesson 1 and input the answers.		
7	Interpret the evaluation feedback.		
8	Log out from the online course.		

TABLE 6 Results of the usability study

Question	Percentage of Users
Is the control for the action visible?	100%
Does the interface allow the user to produce the effect the action has?	100%
Do users succeed in performing this action?	100%
Do users notice that the correct action has been executed successfully?	93%

requirements and the testing activities. This study was also planned from the beginning to give them the opportunity to understand and accept the purpose of the proposed process. The sixth SPI Manifesto principle, "use dynamic and adaptable models as needed," inspired us to combine ISO/IEEE 29110 standard, WCAG, and design patterns in order to integrate accessibility in the context of VSEs that develop software.

The case study included a test process, which integrates three perspectives: (1) automated tools, (2) experts, and (3) users. Therefore, we adopted a holistic approach as suggested by Rubin and Chisnell.[50] The automated tools were used since previous studies show that test automation helps to decrease costs of re-testing and regression testing, as long as they are used in combination with other test methods.[7] The selection of the specific simulators and evaluation tools was based on positive feedback regarding these tools in previous studies.[54] Although visual simulators are not medically accurate, they are considered to be good enough to help developers and testers understand the barriers faced by users with visual disabilities. As for the number of experts, according to Quiñones and Rusu,[51] three is the optimal number of experts, and the number of users exceeds the size suggested by Schmettow.[49] As with any research study, there are some threats to validity which should be considered in regard to the present study. The proposed process was implemented in the development of one web application. The type of web application developed in the present study, an online course, is not necessarily representative of the wide range of web applications. Moreover, although the web application included multimedia educational content, it lacks of complex dynamic features like interaction elements, eg, drag-and-drop or controls. Therefore, future research should examine more complex and dynamic web applications. Nevertheless, the features of the web application developed are comparable to many types of web applications, including web applications of industry, e-government, and social media.

Although we present empirical evidence on the positive effects of accessible design on visually impaired users, the present study is only a first contribution toward including accessibility considerations in the ISO/IEC 29110 standard. As in any other study, there are many possible factors that may have influenced the results. First, recruiting people with visual impairments and computer literacy is of great difficulty. Therefore, previous studies conducting experiments with people with visual impairments tested rather small number of participants, eg, 9 or 11.[55,56] Although we tested a sample of 14 participants, which is in concordance with the 10 ± 2 rule of Hwang and Salvendy,[49] the sample is still rather small for statistical analysis and inference. Therefore, a further study would examine a large, randomly selected sample of participants to improve the accuracy of the findings. More evidence is also required in order to suggest the generalizability of these outcomes to broader populations.

Finally, regarding the ethnography of the participants' sample, due to practical circumstances, the blind users were on average older than the users with blindfolds or simulators (36 years vs 24.5 years). Nevertheless, we do not consider this a serious validity threat, but it would be interesting to have a more balanced sample in future studies.

7 | CONCLUSION

The international standard ISO/IEC 29110 has the potential to positively impact the software industry. Nevertheless, it needs to be integrated and enriched with other proposals such as the accessibility design patterns proposed in this work. This study set out to critically examine ways for improving accessibility using four patterns might be useful while developing software products using the ISO/IEC 29110 standard. Taken together, our results suggest that users with visual disabilities could benefit from the proposed features. These findings contribute in several ways to our goal of integrating accessibility design patterns with SI process and provide a basis for further research. Although this study is focused on users

with visual impairments, it would be interesting in future research to compare effects of accessibility design patterns on both nondisabled users and users with visual impairments rather than focusing on disabled users alone. This comparison will allow us to achieve a deeper understanding of the co-relation between accessibility and usability as suggested by Schmutz et al.[8]

Finally, future research should include an extensive validation of the four proposed accessibility design patterns with experts and more users before publishing them in an online software patterns library.

ACKNOWLEDGEMENT

We want to thank Franklin Jauly and Nicholas Hoekstra for providing requirements for the application from their perspective as blind users. Also, thanks to Santiago Gualotuña and Angel Molina for developing the web application of the case study.

ORCID

Mary Sánchez-Gordón http://orcid.org/0000-0002-5102-1122
Rory V. O'Connor http://orcid.org/0000-0001-9253-0313

REFERENCES

1. Clarke P, O'Connor RV. The situational factors that affect the software development process: towards a comprehensive reference framework. *Info Softw Technol J*. 2012;54(5):433-447. https://doi.org/10.1016/j.infsof.2011.12.003
2. Clarke PM, O'Connor RV, Solan D, et al. Exploring software process variation arising from differences in situational context. *Comm Comp Info Sci J*. 2017;748:29-42. https://doi.org/10.1007/978-3-319-64218-5_3
3. International Organization for Standardization (ISO). ISO/IEC 25000:2014 Systems and software engineering—Systems and Software Quality Requirements and Evaluation (SQuaRE)—guide to SQuaRE. 2014.
4. Sanchez-Gordon S, Luján-Mora S. An ecosystem for corporate training with accessible MOOCs and OERs. In Proceedings of 3rd International Conference on MOOCs, Innovation and Technology in Education (MITE). IEEE, 2015; 123-128. https://doi.org/10.1109/MITE.2015.7375301
5. Bourne RRA, Flaxman SR, Braithwaite T, et al. Magnitude, temporal trends, and projections of the global prevalence of blindness and distance and near vision impairment: a systematic review and meta-analysis. *Lancet Glob Health*. 2017;5(9):e888-e897. https://doi.org/10.1016/S2214-109X(17)30293-0
6. Schmutz S, Sonderegger A, Sauer J. Implementing recommendations from web accessibility guidelines: a comparative study of nondisabled users and users with visual impairments. *Hum Factors* 2017; 59(6): 956–972. https://doi.org/10.1177/0018720817708397
7. Sanchez-Gordon, S., Sánchez-Gordón, M. L., & Luján-Mora, S. Towards engineering process for developing accessible software in small software enterprises. In Proceedings of 11th International Conference on Evaluation of Novel Software Approaches to Software Engineering (ENASE), 2016; 241-246. https://doi.org/10.5220/0005900702410246
8. Schmutz S, Sonderegger A, Sauer J. Implementing recommendations from web accessibility guidelines: would they also provide benefits to nondisabled users. *Hum Factors*. 2016;58(4):611-629. https://doi.org/10.1177/0018720816640962
9. Kleynhans SA, Fourie I. Ensuring accessibility of electronic information resources for visually impaired people. *Library Hi Tech*. 2014;32(2):368-379. https://doi.org/10.1108/LHT-11-2013-0148
10. Sánchez-Gordón ML, O'Connor RV. Understanding the gap between software process practices and actual practice in very small companies. *Softw Qual J*. 2016;24(3):549-570. https://doi.org/10.1007/s11219-015-9282-6
11. Laporte C, Alexandre S, O'Connor R. A software engineering lifecycle standard for very small enterprises. In: O'Connor RV, Baddoo N, Smolander K, Messnarz R. (eds), *Software Process Improvement*. Berlin: Springer-Verlag, 2008; 16: 129-141. https://doi.org/10.1007/978-3-540-85936-9_12
12. Sanchez-Gordon ML, O'Connor RV, Colomo-Palacios R. Evaluating VSEs viewpoint and sentiment towards the ISO/IEC 29110 standard: a two country grounded theory study. In: Rout T, O'Connor RV and Dorling A. (eds), *Software Process Improvement and Capability Determination*. Berlin: Springer-Verlag, 2015; 526:114-127. https://doi.org/10.1007/978-3-319-19860-6_10
13. Mesquida AL, Mas A. A project management improvement program according to ISO/IEC 29110 and PMBOK. *J Softw: Evol Process*. 2014; 26(9):846-854. https://doi.org/10.1002/smr.1665
14. Larrucea X, O'Connor RV, Colomo-Palacios R, Laporte CY. Software process improvement in very small organizations. *IEEE Softw*. 2016;33(2):85-89. https://doi.org/10.1109/MS.2016.42
15. O'Connor RV, Laporte CY. An innovative approach to the development of an international software process lifecycle standard for very small entities. *Int J Info Technol Syst Approach*. 2014;7(1):1-22. https://doi.org/10.4018/ijitsa.2014010101
16. Larrucea X, Santamaría I, Colomo-Palacios R. Assessing ISO/IEC29110 by means of ITMark: results from an experience factory. *J Softw: Evol Process*. 2016;28(11):969-980. https://doi.org/10.1002/smr.1795
17. De Branco RG, Cagnin MI, Paiva DMB. AccTrace: accessibility in phases of requirements engineering, design, and coding software. In Proceedings of the IEEE 4th International Conference on Computational Science and Its Applications (ICCSA), 2014; 225-228. https://doi.org/10.1109/ICCSA.2014.51
18. O'Connor RV, Laporte CY. The evolution of the ISO/IEC 29110 set of standards and guides. *Int J Info Technol Syst Approach*. 2017;10(1):1-21. https://doi.org/10.4018/IJITSA.2017010101
19. Moreno-Campos E, Sanchez-Gordón ML, Colomo-Palacios R, De Amescua Seco A. Towards measuring the impact of the ISO/IEC 29110 standard: a systematic review. *Comm Comp Info Sci*. 2014;425:1-12. https://doi.org/10.1007/978-3-662-43896-1_1
20. International Organization for Standardization (ISO). ISO/IEC TR 29110-1 Systems and software engineering—lifecycle profiles for very small entities (VSEs)—Part 1: Overview. 2016.
21. Laporte CY, O'Connor RV, Fanmuy G. International systems and software engineering standards for very small entities. *CrossTalk, J Defense Softw Eng*. 2013;26:28-33.

22. International Organization for Standardization (ISO). ISO/IEC TR 29110-5-1-1 Software engineering—lifecycle profiles for very small entities (VSEs)—Part 5-1-1: Management and engineering guide: generic profile group: entry profile. 2012.
23. International Organization for Standardization (ISO). ISO/IEC TR 29110-5-1-2 Software engineering—lifecycle profiles for very small entities (VSEs)—Part 5-1-2: Management and engineering guide: Generic profile group: Basic profile. 2011.
24. International Organization for Standardization (ISO). ISO/IEC TR 29110-5-1-3, software engineering—lifecycle profiles for very small entities (VSEs)—Part 5-1-3: management and engineering guide: generic profile group: intermediate profile. 2017.
25. International Organization for Standardization (ISO). ISO/IEC TR 29110-5-1-4, software engineering—lifecycle profiles for very small entities (VSEs)—Part 5-1-3: management and engineering guide: generic profile group: advanced profile. 2018.
26. International Organization for Standardization (ISO). ISO/IEC TR 29110-3, Software engineering—lifecycle profiles for very small entities (VSEs)—Part 3: assessment guide. 2015.
27. Takeuchi M, Kohtake N, Shirasaka S, Koishi Y, Shioya K. Report on an assessment experience based on ISO/IEC 29110. *J Softw: Evol Process*. 2014;26(3):306-312. https://doi.org/10.1002/smr.1591
28. Cervantes H, Kazman R. *Designing Software Architectures: A Practical Approach. The SEI Series in Software Engineering*. Boston: Assison-Wesley; 2016.
29. Gamma E, Helm R, Johnson R, Vlissides J. *Design Patterns: Elements of Reusable Object-Oriented Software*. *Addison Wesley Professional Computer Series*. Indianapolis: Oxford University Press; 2009.
30. Kruschitz C, Hitz M. Human-computer interaction design patterns: structure, methods, and tools. *Int J Adv Softw*. 2010;3(1):225-237.
31. Seffah A. Patterns of HCI design and HCI design of patterns: bridging HCI design and model driven software engineering. *Hum-Comp Interact Ser Switzerland*: Springer International Publishing. 2015. https://doi.org/10.1007/978-3-319-15687-3_10
32. Borchers JO. A pattern approach to interaction design. In: Gill S. (ed) *Cognition, Communication and Interaction*. *Human-Computer Interaction Series*. London: Springer, 2008; 114-131. https://doi.org/10.1007/978-1-84628-927-9_7
33. Nunes NJ. Representing user-interface patterns in UML. In *International Conference on Object-Oriented Information Systems*. Berlin: Springer, 2003; 142-151. https://doi.org/10.1007/978-3-540-45242-3_14
34. Henninger S, Corrêa V. Software pattern communities: current practices and challenges. In Proceedings of the 14th Conference on Pattern Languages of Programs. ACM. 2007: 14. https://doi.org/10.1145/1772070.1772087
35. Kruschitz C, Hitz M. Are human-computer interaction design patterns really used? In Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries, ACM. 2010: 711-714. https://doi.org/10.1145/1868914.1869011
36. United Nations (UN). Global status report on disability and deployment. 2015.
37. World Health Organization (WHO). World disability report. 2011.
38. Colour Blind Awareness (CBA). Colour blindness. 2018.
39. Eye Health Web (EHW). Blurred vision. 2018.
40. U.S. Department of Education Office of Civil Rights (OCR). Resolution agreement South Carolina Technical College System. 2013. http://www2.ed.gov/about/offices/list/ocr/docs/investigations/11116002-b.html
41. Sanchez-Gordon S, Luján-Mora S. Design, implementation and evaluation of MOOCs to improve inclusion of diverse learners, In: Mendoza-Gonzalez R (ed.) *User-Centered Design Strategies for Massive Open Online Courses (MOOCs)*, Hershey: IGI Global, 2016; 115-141. https://doi.org/10.4018/978-1-4666-9743-0.ch008
42. World Wide Consortium (W3C). Web content accessibility guidelines WCAG 2.1.2018. https://www.w3.org/TR/WCAG21/
43. Fortin M. *Sim Daltonism*. 2014. https://michelf.ca/projects/sim-daltonism/
44. World Wide Web Consortium (W3C). Web accessibility evaluation tools list. 2016. https://www.w3.org/WAI/ER/tools/
45. W3C. Markup validation service. 2013. http://validator.w3.org
46. World Wide Web Consortium (W3C). Notes on using ARIA in HTML. 2015. http://www.w3.org/TR/aria-in-html/
47. WebAIM. WCAG 2.0 checklist for HTML documents. 2013. https://webaim.org/standards/wcag/checklist
48. Nielsen J, Molich R. Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '90)*. In: Carrasco J and Whiteside J (eds.). ACM, New York, 2010; 249-256. https://doi.org/10.1145/97243.97281
49. Schmettow M. Sample size in usability studies. *Comm ACM J*. 2012;55(4):64-70. https://doi.org/10.1145/2133806.2133824
50. Rubin J, Chisnell D. *Handbook of Usability Testing. How to Plan, Design, and Conduct Effective Tests. Second Edition*. Boston: Willey Publisher; 2008.
51. Quiñones D, Rusu C. How to develop usability heuristics: a systematic literature review. *Comp Standards Interf*. 2017;53:89-122. https://doi.org/10.1016/j.csi.2017.03.009
52. Pascual A, Ribera M, Granollers T, Coiduras JL. Impact of accessibility barriers on the mood of blind, low-vision and sighted users. In 5th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion. *DSAI*. 2014;27:431-440. https://doi.org/10.1016/j.procs.2014.02.047
53. Pries-Heje J, Johansen J. *SPI Manifesto*. 2010. http://www.iscn.com/Images/SPI_Manifesto_A.1.2.2010.pdf
54. Sanchez-Gordon S, Luján-Mora S. A method for accessibility testing of web applications in agile environments. In proceedings of the 7th World Congress for Software Quality (WCSQ), 2017; 1-8.
55. Aizpurua A, Arrue M, Vigo M. Prejudices, memories, expectations and confidence influence experienced accessibility on the web. *Comp Human Behav*. 2015;51:152-160. https://doi.org/10.1016/j.chb.2015.04.035
56. Rømen D, Svanæs D. Validating WCAG Versions 1.0 and 2.0 through usability testing with disabled users. *Univ Access Info Soc*. 2012;11(4):375-385. https://doi.org/10.1007/s10209-011-0259-3

How to cite this article: Sanchez-Gordon S, Sánchez-Gordón M, Yilmaz M, O'Connor RV. Integration of accessibility design patterns with the software implementation process of ISO/IEC 29110. *J Softw Evol Proc*. 2018;e1987. https://doi.org/10.1002/smr.1987