# A Conceptual Model For Web Accessibility Requirements In Agile Development

Darliane Miranda†
NOVA LINCS
NOVA University of Lisbon
Lisbon, Portugal
dg.miranda@fct.unl.pt

João Araújo
NOVA LINCS
NOVA University of Lisbon
Lisbon, Portugal
p191@fct.unl.pt

Grischa Liebel
School of Technology
Reykjavik University
Reykjavik, Iceland
grischal@ru.is

## ABSTRACT

Accessibility is the practice of making content and functionality accessible to all users, regardless of their abilities. Although accessibility is a highly relevant quality attribute, it is often treated as an afterthought in software development, unfortunately excluding people with disabilities from using many web-based systems. Specifically in agile development, sprints focus on new features and quality attributes, such as accessibility, are often not considered sufficiently. In these cases, using conceptual models to understand and analyze requirements that developers have formulated as a set of related user stories is a research opportunity. To increase agile professionals' focus on accessibility, we built a conceptual model for web accessibility, identifying artifacts and concepts used in agile development to specify accessibility. We discuss how this model can be used as a guide to better integrate accessibility considerations into agile software development. Researchers can use the result to define resources that are not currently covered or improve underutilized practices. We plan to use the conceptual model in the next steps to adapt existing agile artifacts and create support tools for web accessibility in agile development.

## CCS CONCEPTS

• Software and its engineering • Software creation and management • Designing software • Requirements analysis

## KEYWORDS

Accessibility Requirements, Agile Development, Conceptual Model, Requirements Engineering



MO2RE 2024, April 16, 2024, Lisbon, Portugal

ACM ISBN 979-8-4007-0569-4/24/04.
https://doi.org/10.1145/3643666.3648580

## 1 Introduction

According to the World Health Organization [1], over a billion people are estimated to live with some form of disability, which corresponds to about 16% of the world's population. Furthermore, according to the International Telecommunication Union [2], about 3.2 billion people use the internet. Considering this substantial number of users, web-based systems should be designed to provide access to information and interaction for all people, including people with disabilities. However, issues involving web accessibility specification, implementation, and evaluation are common [3]. Web accessibility combines programming, design, and technology to build a barrier-free internet that allows all users to understand, learn, navigate and fully interact with the web.

The Web Accessibility Initiative (WAI) [4] is the worldwide inducer of this field, which ensures that web pages and their contents include usability and accessibility so that any internet user can browse the web regardless of their physical, intellectual, or sensory limitations. Since 1999, the WAI has issued standardized and internationally recognized guidelines to create more accessible web content and pages. These technical recommendations, such as the Web Content Accessibility Guidelines (WCAG), are specially designed for web developers and designers. To produce high-quality software that meets customer expectations and demands, software engineers commonly apply continuous software quality control in the context of agile methods [5]. However, ensuring accessibility throughout the development process is challenging [6,7]. Despite its relevance, most guidelines are extensive documents that can be difficult to interpret if users do not have sufficient knowledge on the domain [8].

Recent research seeks to integrate accessibility into agile software development by selecting the best techniques for accessibility testing from a cost-benefit point of view [9]. Among other alternatives to improve accessibility practices within agile teams (such as frameworks, tools, methods, etc.), the use of conceptual models may be a viable alternative since they can serve as a guiding framework that supports the principles of agile methodologies while emphasizing the importance of inclusive design and accessibility considerations. Conceptual modelling in agile development is an underexplored area and may be solution

to accessibility requirements challenges [10]. Also, it contributes to the creation of software that is not only functional and adaptable, but also accessible to a diverse user base.

Therefore, in order to better consider and increase agile practitioners focus on accessibility, we propose a conceptual model to improve the specification of accessibility requirements based on the conceptualization of accessibility as an inclusive and social construct. This study offers a comprehensive overview of existing approaches, their characteristics, applicability and possible limitations. Because the WCAG accessibility guidelines are the reference in this domain, we collect the elements and components related to accessibility specification, implementation, and testing and bring these elements together to create a knowledge base. In the next steps, we will validate the resulting model and use it to adapt specific agile practices, such as accessibility sprints.

## 2 Accessibility Requirements

Governments around the world created legislation to ensure equality of access and participation in the information society [11]. To fulfil web accessibility requirements, IT professionals usually rely on international guidelines that provide a wide range of recommendations for making web content more accessible. An example is WCAG, primarily intended for web content developers (page authors, site designers, etc.), web authoring tool developers, and others who want or need a standard for web accessibility.

Web accessibility relies on several components that work together: the web content, user agents, and authoring tools. Web content generally refers to the information in a web page or application, including text, images, and sound, alongside code or markup that defines the structure and presentation. User agents include browsers, extensions, media players, readers, and other applications that render web content, i.e., software that people use to access web content. Authoring tools refer to software or services that people use to produce web content, for example, Content Management Systems (CMS) and code editors.

Despite the relevance of the accessibility, a recent survey indicated that professionals struggle in understanding or even finding the necessary information in the available international guidelines, facing difficulties to find specific information and, sometimes, dealing with outdated content [8]. Besides accessibility guidelines, there are relevant studies that propose methods, approaches, and processes for promoting the dissemination of accessibility requirements among IT professionals, but the specification of these requirements still needs more attention from the practitioners and industry [12].

### 2.1 Accessibility in Agile Development

Although the adoption of agile methodologies intends to deliver high-quality products, many organizations still adopt a waterfall approach to accessibility, leaving it to the end of the project lifecycle [9]. Some researchers are concerned with the treatment of non -functional requirements (NFRs) and suggested approaches to deal with its challenges [13-14]. However, approaching accessibility requirements specifically requires more attention since it demands a certain level of knowledge about the domain, implementation, testing techniques, and its integration with assistive technologies. Some of the key integration points for accessibility in agile development include iterative testing, design and architecture, and incorporation of accessibility in the development team's Definition of Done (DoD) [9]. Although the DoD can vary from one team to another, depending on stakeholders' requirements and development team preferences, all product increments should conform to accessibility guidelines, such as WCAG, to be considered done.

Recent research on artifacts employed within the framework of agile development for delineating web accessibility requirements reveals a predominant emphasis on evaluation methods and tools [15] geared towards assessing a system's compliance with established standards and guidelines. However, it is possible to verify a gap in approaches that address accessibility requirements through existing agile artifacts (e.g. user stories and personas) in the absence of direct involvement from accessibility experts. Furthermore, there is a lack of research (see more in section 3) that shows, through models, the fundamental components in the domain of web accessibility, their interrelationships, and possible strategies for their integration into agile methodologies.

### 2.2 A conceptual model for web accessibility in agile software development

Conceptual modeling involves creating abstract representations that capture the fundamental concepts, entities, relationships, and behaviors of the system or domain. It focuses on what is essential and omits unnecessary details [16]. According to John Mylopoulos [16], some key aspects of conceptual modeling include: (a) understanding and modeling the problem domain, which represents the real-world aspects and requirements that the system aims to address; (b) understanding user needs and expectations is essential for creating models that reflect the true requirements of the system; (c) conceptual modeling is often an iterative process which involves refining and revising models based on feedback, changing requirements, and evolving understanding of the problem domain and (d) have ontological commitment, i.e., models should reflect a shared understanding of the entities, relationships, and constraints in the problem domain.

Overall, conceptual modeling emphasizes the importance of creating clear and abstract representations that capture the essence of a system or domain. These models serve as a foundation for effective communication, understanding, and development throughout the software engineering process.

When it comes to agile, while visual models have proven their effectiveness in supporting requirements engineering, their use is not advocated by agile proponents [10]. Recently, studies have suggested that the use of conceptual models to understand and analyze requirements that developers have formulated as a set of related user stories (e.g., a theme or epic in Scrum) is a research opportunity [17]. Considering accessibility to this scenario, it is possible to align principles of inclusivity, user-centered design,

and efficient development practices within the agile framework, by:

- Empowering development teams: the model may act as a knowledge base to support developers, designers, and other team members in implementing inclusive design features.
- Compliance with accessibility standards: many conceptual models align with established accessibility standards, such as WCAG. Adhering to these standards is crucial for meeting legal and regulatory requirements related to accessibility.
- Efficient testing and validation: having a conceptual model may help teams plan and conduct more efficient accessibility testing. It provides a framework for creating test cases and validating that the software meets the defined accessibility criteria.

To build our conceptual model, we adapt steps suggested by [25], which include:

1. Determine the domain and scope of the model: the conceptual model for the accessibility domain was designed as a reference for agile teams to consistently apply accessibility practices across sprints and iterations.
2. Identify key concepts: the concepts were based on the WCAG and represent the essential elements of the domain.
3. Define relationships: determine how the identified concepts are related to each other and specify the connections, dependencies, or interactions between concepts.
4. Create a formal representation: choose a formal representation language or tool to depict the conceptual model. In our model we use a UML (Unified Modeling Language) class diagram.
5. Validate and refine: validate the conceptual model by comparing it against real-world scenarios or data. Seek feedback from domain experts to refine and improve the model. Although this step was not included in this paper, it is part of future work.

In summary, a conceptual model for accessibility in agile software development may serve as a guiding framework that supports the principles of agile methodologies while emphasizing the importance of inclusive design and accessibility considerations. It contributes to the creation of software that is not only functional and adaptive but also accessible to a diverse user base.

## 3 Related Work

Recent studies were published to state the practices of web accessibility requirements considering the agile [18-24]. Silva et. al. [18] propose a conceptual model of the software development process that proposes the inclusion of accessibility throughout the software life cycle. The presented model combines user-centered and agile development methodologies (UCASD - User-Centered Agile Software Development) and encompasses five major phases (analysis, design, coding, development and maintenance). It provides a theoretical perspective, merging into a single document a set of acknowledgements from the existing literature on e-accessibility.

Sane, Parth [19] attempts to map the current scene of the software engineering initiatives dedicated to enhancing accessibility through continuous integration, along with the challenges hindering its widespread adoption. The paper also introduces essential diagrams to illustrate processes for incorporating accessibility testing procedures into the software development lifecycle using continuous integration.

Fathauer and Rao [20] implemented an agile development process that incorporates the expertise of an accessibility specialist. The paper outlines the agile methodology and shares insights from the design and development of an accessible software system. Also highlights some of the recommended accessibility enhancements aimed at meeting established accessibility standards.

Romero-Chacón, Víctor, et al. [21] aimed to incorporate accessibility into Inclutec1 products using an adaptation of the SCRUM methodology. This was achieved by integrating accessibility tasks into the development process, where the phases of accessibility tests, accessibility corrections and accessibility reviews were added to sprints aimed to achieve accessible products.

Stray, V., Bai, A., Sverdrup, N., and Mork, H. [22] conducted a case study in a Norwegian software company to understand how accessibility testing can be better integrated into the daily routine of agile projects. The authors investigated three accessibility testing tools: automatic checker, simulation glasses, and a dyslexia simulator. The results show that all three tools are suitable for agile projects.

Bai, A., Mork, H. C., & Stray, V. [23] introduce an assessment of nine accessibility testing approaches tailored for integration into an agile software development process. Through a cost-benefit analysis, was identify an optimal combination of these methods, considering both cost and the identification of issues. Ultimately, was outlined the integration of accessibility testing methods into an agile process through the utilization of the agile accessibility spiral.

Sanchez-Gordon1 and Luján-Mora [24], propose the adoption of a holistic approach to accessibility testing in agile environments, incorporating automated tools, simulators, expert-based testing, and user-based testing. The proposed method aligns with the five stages outlined 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. Within each of these stages, the method delineates precise tasks tailored for conducting accessibility testing of web applications within the agile development context.

The mentioned studies represent a progression in the research on web accessibility within agile development. However, the quantity of studies and their practical applicability remain areas in need of improvement. Among the cited studies, only one

[1] Inclutec is an interest group of the Costa Rica Institute of Technology (TEC), in the development of the SICID platform for the Council of Persons with Disabilities (CONAPDIS) in Costa Rica.

highlights the utilization of a conceptual model, lacking both a visual representation and a detailed description of the fundamental components endorsed by WCAG for creating accessible systems. Additionally, there is a lack of approaches or frameworks that could adapt artifacts already used in an agile context, such as user stories and personas, to specify accessibility requirements without inclusion of an accessibility specialist in the process. Hence, this work aims to fill this gap by combining the advantages of using conceptual models with key components of the web accessibility domain.

## 4 A Conceptual Model For Web Accessibility Requirements

To proactively identify and mitigate accessibility issues from the beginning and throughout the software development life cycle and support professionals in the elicitation of accessibility requirements, we created a conceptual model for web accessibility, as illustrated in Figure 1. The development process was based on the methodologies proposed in [25]. This model serves to depict and explain web accessibility components and their interconnections. Its constituents were derived from the WAI accessibility guidelines, encompassing elements pertinent to agile development. The conceptual model is represented as a UML class diagram and serves to establish the correlation between *Web Accessibility* and the development of *Web-Based System* within an agile context. In this context, *Web Accessibility* is regulated by *Norm*, which can encompass *Standard*, *Law*, and Organizational policy. Legal obligations, as defined by *Law*, can vary depending on the country and the types of web content. Subsequently, we will provide a more detailed description of the model's constituent elements.

**Component.** An identifiable part of software development that provides a particular function or group of related functions. Several different components of web development and interaction must work together for the web to be accessible to people with disabilities. These *Component* include *Content*, *Web Browser*, and *Authoring tool*, and interact with *Evaluation tools* and *Assistive Technology*.

**Content.** Data created for a specific purpose. It is a part of the system that people (with and without disabilities) will interact with, whether through a website, social media, etc. It is the result of the product presented to users. This component is at the heart of web accessibility design as it guides how the system will be developed and consumed by the end-user.

**Web browsers.** Tools allowing users to interact with HTML documents and must have an interface that is accessible. In an accessibility context, it is considered a user agent along with browser extensions, media players, readers, and other applications that render web content. Like any other system, a web browser must be compatible with *Assistive Technology*.

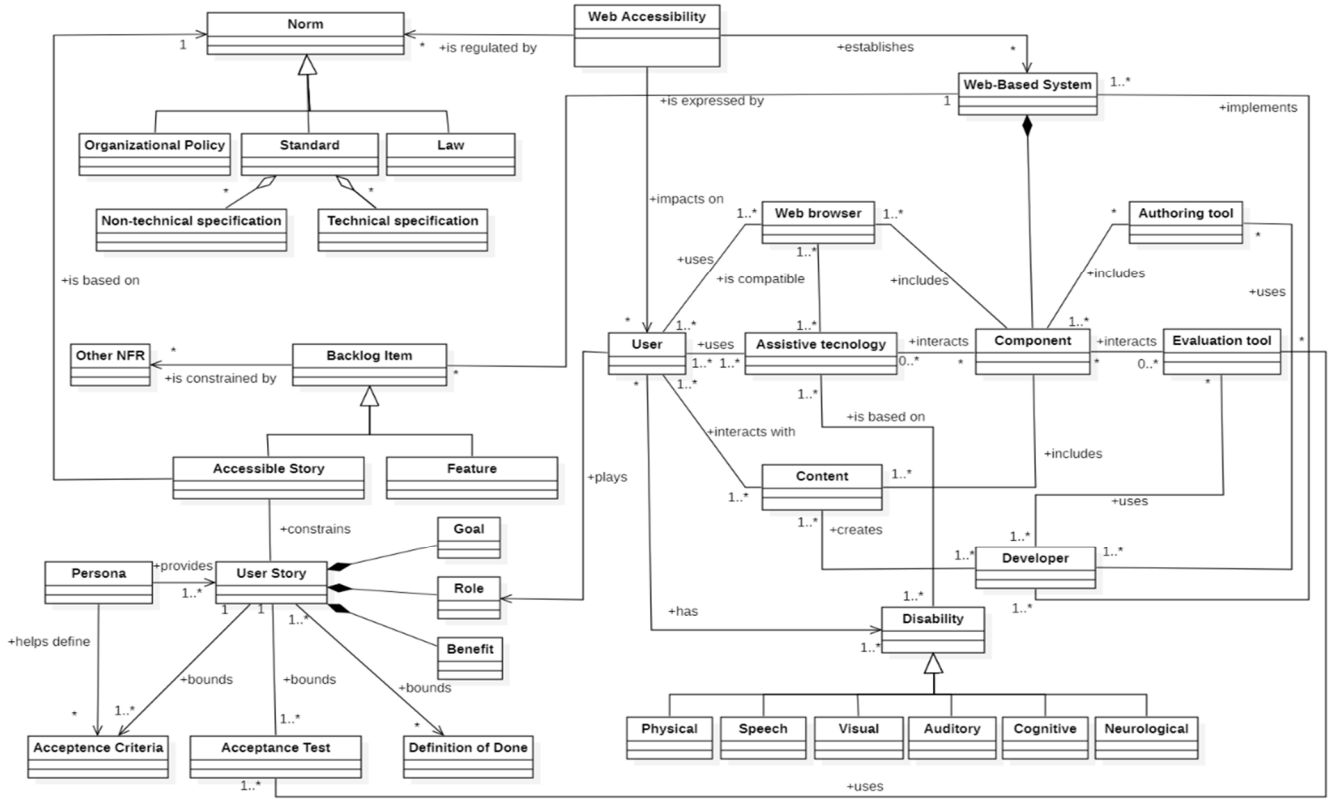


**Figure 1: Conceptual model for web accessibility**

**Assistive technology.** In the web context, refers to hardware or software used to increase, maintain or improve functional capabilities of individuals with disabilities. Examples include screen readers, which audibly convey web page content for those unable to read text, and voice recognition software alongside selection switches for those facing challenges using a traditional keyboard or mouse.

**Authoring tool.** Refer to software and services utilized by *Developer* (which can include web developers, designers, writers, and others) for the creation of web content. A few examples are eLearning authoring tools (e.g., Elucidat), software for generating/managing websites (e.g., WordPress), and web application frameworks, IDEs, and SDKs.

**Evaluation tools**. Software or online services that help to determine if web content meets accessibility standards. Tools can provide fully automated checks and help with manual review. Although these tools check all accessibility aspects automatically, human judgment is required to provide real perception of the accessibility of a web system.

**User.** This is the central component of our model, as the development of accessible systems must be guided by the perception of the end-user and their needs. The concept of *Web Accessibility* directly impacts the *User*, who may or may not have any of these types of *Disability*, which in our model are represented by: *Physical, Speech, Visual, Auditory, Cognitive,* and *Neurological* [1]. Users who have any disability can use *Assistive Technology* to access *Content* presented on the web and, consequently, interact with one or more *Component* of a W*eb-based system*. The user will need to access web content through a *Web Browser* that needs to be developed in an accessible and compatible way with *Assistive Technology*, thus allowing the user to interact with the *Content.*

**Developer**. Can be represented by designers, developers, authors, etc., including those with disabilities and users who contribute with content. Developers are responsible for implementing *Web-Based System*, creating *Content*, using *Authoring Tool* and *Evaluation Tool*. These tools may or may not be used by developers, depending on the type of developed systems and the necessity for an evaluation tool.

**Backlog Item**. In our model, web based system behavior is expressed as *Backlog Item*, where *Accessible Story* and *Feature* are a kind of *Backlog Item* and also may be constrained by *Other NFRs* (such as privacy, security, safety, etc.). Since other functionalities of the system are represented by Feature, we are focused on the Accessible Story, defined as a work item contained in the team's iteration/sprint backlog. The Accessible Story should be based on Norms to help developers understand user needs according to the type of disability and implement them. Since we are considering in this model the agile context, requirements are often expressed as User Stories.

**User Story**. Considering the well-known template of *User Story* [26] (As a [role], I want [a goal] so that I can [achieve a specific benefit]), we decided to use this artifact to express accessibility requirements. For an accessible user story, we envision user stories that consider the specificity of each user and their possible shortcomings at the time of user story creation. An example of a user story for accessibility is: *As a screen reader user, I want to hear the text equivalent for each image button so that I can understand their functionalities*. In our model, we have represented the *Acceptance Test* as an artifact distinct from the *User Story*, since it will confirm that the story has been implemented correctly (the use of *Evaluation Tool* can assist in that task). Every *User Story* has at least one *Acceptance Test*, which must be passed before the *Accessible Story* can be considered finished. Similarly, a *Definition of Done* helps preventing delivery of systems without appropriate accessibility features. *Acceptance Criteria* are also an essential part of user stories, since they provide a list of conditions that a web-based system must meet to be considered accessible and can also be defined by *Persona* which describes a potential user of the system to be developed and it represents a larger part of the target group.

## 5 Application and discussion of the conceptual model

The goal of this model is to make explicit the impact of accessibility on agile software development. In turn, this helps to facilitate communication, improve decision making, and improve the resulting documentation of accessible web systems. To highlight the application of this model, we consider an example case using the Scrum method, where each sprint involves a complete front-to-back lifecycle, usually including the delivery of a partial subset of features to solicit customer feedback. Figure 2 represents activities related to accessibility concerning planning, development, and testing, based on our conceptual model. To ensure accessibility, these activities should be considered in each sprint or at least in dedicated accessibility sprints. The sprint review and retrospective should reflect and continuously help improve accessibility-related skills.

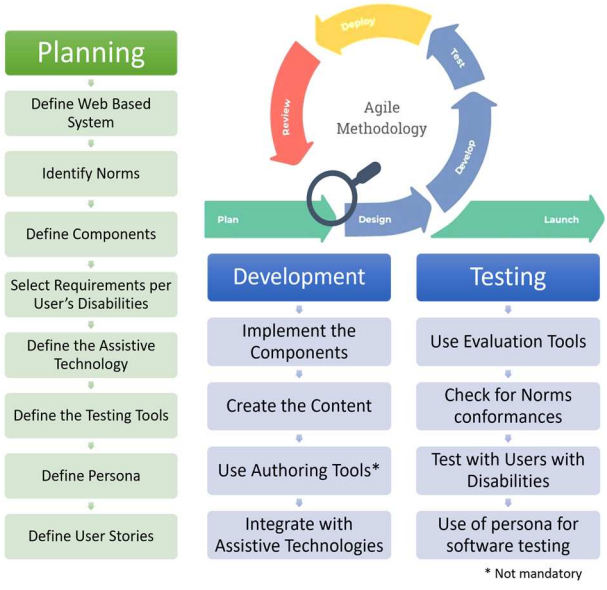


**Figure 2: The proposed conceptual model applied to Scrum**

This model can be tailored and applied according to the organization's goals and the need of each project, but must consider: i) identifying which norms will be followed; ii) aligning requirements and obligations according to the type of system to be developed (e.g. websites, or authoring tools); iii) defining the assistive technologies used during development and testing; iv) defining the accessibility testing tools used and the type of test to be conducted (e.g., online testing tools, tests with users with disabilities); v) making explicit the requirements according to the types of disability; and vi) defining user stories based on the essential components and elements of the model.

To understand how our conceptual model can help professionals in the agile context to specify accessibility requirements, we chose a practical hotel booking example to demonstrate its application. For a better illustration, we used a specification of a persona for accessibility, represented by Susan, archetypical of a visually impaired persona, representing blind users (see Figure 3).

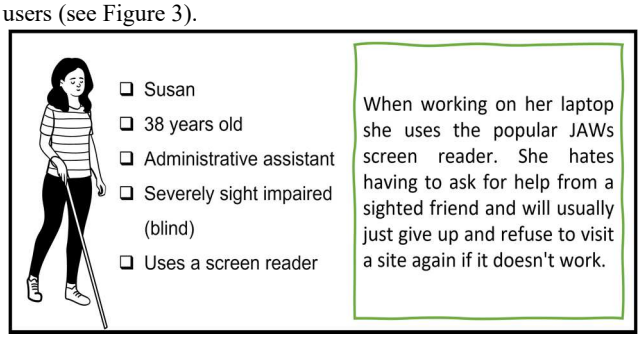


**Figure 3: Persona card of Susan, a blind person**

When accessing a particular hotel reservation website, Susan's first perception will be the compatibility of the web system with the screen reader she uses, the JAWs. When accessing web content, the screen reader uses structural information of a web page, such as headers, column and row headers in tables, list items, links, form controls, and more that allow to navigate the page and access information effectively. If the site fails to implement some of these elements, Susan will not be able to use the website without help and will likely not return to the web page. Thus, assistive technologies need to interact with the site's components, such as the content presented. The same applies for using the chosen web browser. When developing a booking site, the developer must use a system that correctly interacts with the hotel booking system components. The developer can use an authoring tool to develop the reservation system, such as, for example, WordPress. Using a CMS, the developer can/will have to make an adjustment related to accessibility, e.g., by checking accessibility compliance using evaluation tools such as Achecker[2]. These tools raise critical errors and propose solutions, e.g., warning if image captions are missing. The accessibility requirements will be expressed through user stories, based on the standards, to identify which requirements will be part of the project. A possible user story for this project would be: As a blind user, I want to choose a specific room so that I can conclude my reservation. A possible Acceptance Criteria could be: The login function (e.g., button) must have an accessible label (e.g., using 'label for=', 'aria-label=', or other semantically appropriate programmatic association). The Definition of Done should check if the documentation is updated (standards), unit tests were written and passed, and acceptance criteria for the user story fulfilled. In case the reservation system belongs to a government unit, specific legislation might exist that requires companies to deliver a product accessible to all users, including users with physical, hearing, speech, visual, cognitive, or neurological disabilities. Therefore, mandatory requirements might exist for the system to be implemented. Applying the activities derived from our conceptual model, the example system can be developed in a way accessible to users like Susan.

[2] https://achecks.org/checker/index.php

## 6 Concluding remarks and future work

In this paper, we present a conceptual model for web accessibility in agile development. Based on the literature, the concern about the specification of accessibility requirements is increasing, but there are few solutions in the agile development targeting the web accessibility requirements. Also, the related work shows that most of studies that approach web accessibility are focusing on frameworks and tools, and there are few dedicated to the use of models in their applicability.

The conceptual model presented in this paper provides a clear overview of the domain, consisting of the concepts essential to web accessibility and their relationships. The elements of the model are derived from the WAI accessibility guidelines, covering aspects relevant to agile development. Presented as a UML class diagram, the conceptual model aims to establish the connection between web accessibility and the development of web-based systems within an agile context. We exemplified the use of the model through hypothetical practices in Scrum and through an example scenario.

We expect that the contributions of this paper can help those who want to have a clear and more objective view of accessibility in web systems and can serve as a guide for agile practitioners. For instance, agile practitioners can use the results of this research to support their quality planning, implementing, and testing activities, tailoring existing artifacts or tools to the needs of each project.

However, to this end, further work on the conceptual model is necessary. First, we aim to evaluate the model through a survey with accessibility specialists and researchers knowledgeable in conceptual modelling. Following this evaluation, we also plan to produce artifacts tailored to existing agile methods, such as an accessibility catalog, user histories (in conformance to the WCAG), and a conflict checklist, available through a web tool.

We will do so using focus groups with experts in agile development and accessibility.

## ACKNOWLEDGMENTS

This work is supported by NOVA LINCS (UIDB/04516/2020) with the financial support of FCT.IP.

## REFERENCES

[1] WHO. Disability and health, https://www.who.int/news-room/fact-sheets/detail/disability-and-health, last accessed 2022/10/15.

[2] International Telecommunication Union (ITU). Statistics, https://www.itu.int/en/ITU-D/Statistics/Pages/stat/default.aspx, last accessed 2022/10/15.

[3] Bi, T., Xia, X., Lo, D., Grundy, J., Zimmermann, T., and Ford, D.,. 2022. *Accessibility in Software Practice: A Practitioner's Perspective*. ACM Trans. Softw. Eng. Methodol.

[4] Rutter, Richard, et al.. 2007. *Web accessibility: Web standards and regulatory compliance.* Apress.

[5] Arcos-Medina, Gloria, and David Mauricio. 2019. *Aspects of software quality applied to the process of agile software development: a systematic literature review*. International Journal of System Assurance Engineering and Management 10, 867-897.

[6] Pellegrini, F., Anjos, M., Florentin, F., Ribeiro, B., Correia, W., and Quintino, J., 2020. *How to prioritize accessibility in agile projects*. Advances in Usability and User Experience: Proceedings of the AHFE, International Conferences on Usability & User Experience, and Human Factors and Assistive Technology. Springer.

[7] Luján-Mora, Sergio, and Firas Masri. 2012. *Integration of web accessibility into agile methods*. In: 14th International Conference on Enterprise Information Systems (ICEIS). p. 123-127.

[8] Miranda, Darliane and Araujo, João. 2022. *Studying Industry Practices of Accessibility Requirements in Agile Development*. In: 37th Symposium On Applied Computing (SAC).

[9] Bai, Aleksander, Heidi Mork, and Viktoria Stray. 2018. *How agile teams regard and practice universal design during software development*. Stud. Health Technol. Inform, 256, 171-184.

[10] Gupta, Abhimanyu, Geert Poels, and Palash Bera. 2022. *Using Conceptual Models in Agile Software Development: A Possible Solution to Requirements Engineering Challenges in Agile Projects*. IEEE Access 10: 119745-119766.

[11] Moreno, Lourdes, and Paloma Martinez. 2019. *The harmonization of accessibility standards for public policies*. Computer, 57-66.

[12] Paiva, D. M. B., Freire, A. P., and de Mattos Fortes, R. P. 2020. *Accessibility and Software Engineering Processes: A Systematic Literature Review*. Journal of Systems and Software, 171.

[13] Heikkilä, V. T., Damian, D., Lassenius, C., and Paasivaara, M. 2015. *A mapping study on requirements engineering in agile software development*. In: 41st Euromicro conference on software engineering and advanced applications. IEEE,p. 199-207.

[14] Schön, E. M., Thomaschewski, J., and Escalona, M. J. 2017. *Agile Requirements Engineering: A systematic literature review*. Computer Standards & Interfaces, 49: 79-91.

[15] Sane, Parth. 2021. *A Brief Survey of Current Software Engineering Practices in Continuous Integration and Automated Accessibility Testing*. In 6th International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET). IEEE.

[16] Mylopoulos, John. 1992. *Conceptual modelling and Telos*. Conceptual modelling, databases, and CASE: An integrated view of information system development, p. 49-68.

[17] Amna, Anis R., and Geert Poels. 2022. *Ambiguity in user stories: A systematic literature review*. Information and Software Technology 145: 106824.

[18] Silva, J. S. E., Gonçalves, R., Branco, F., Pereira, A., Au-Yong-Oliveira, M., and Martins, J. 2019. *Accessible software development: a conceptual model proposal*. Universal Access in the Information Society, 18, 703-716.

[19] Sane, Parth. 2021. *A brief survey of current software engineering practices in continuous integration and automated accessibility testing*. Sixth International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET). IEEE.

[20] Fathauer, Laura; Dhananjai M. Rao. 2019. *Accessibility in an educational software system: Experiences and Design Tips*. IEEE Frontiers in Education Conference (FIE).

[21] Romero-Chacón, V., Muir-Camacho, H., Rodríguez-González, J., Gómez-Blanco, A., and Chacón-Rivas, M. 2019. *Adapting SCRUM methodology to develop accessible web sites*. In 2019 International Conference on Inclusive Technologies and Education (CONTIE), pp. 112-1124. IEEE.

[22] Stray, V., Bai, A., Sverdrup, N., and Mork, H. 2019. *Empowering agile project members with accessibility testing tools: a case study*. In International Conference on Agile Software Development. pp. 86-101. Springer.

[23] Bai, A., Mork, H. C., and Stray, V. 2017. *A cost-benefit analysis of accessibility testing in agile software development: results from a multiple case study*. International Journal on Advances in Software, 10.1&2: 96-107.

[24] Sanchez-Gordon, S., and Luján-Mora, S. 2017. *A method for accessibility testing of web applications in agile environments*. In 7th World Congress for Software Quality (WCSQ).

[25] Noy, N. F., & McGuinness, D. L. 2001. *Ontology development 101: A guide to creating your first ontology*.

[26] Cohn, M. 2004. *User stories applied: For agile software development*. Addison-Wesley Professional.