Date of publication MAR-31, 2025, date of current version SEP-09, 2024.
www.computingonline.net / computing@computingonline.net

*Print ISSN 1727-6209*
*Online ISSN 2312-5381*
*DOI 10.47839/ijc.24.1.3891*

# Model of the Continuous Accessibility Process in the Reality of Commercial Web Development

**OLEKSANDR GURA, OLEKSANDR GURA JR.**
Department of Psychology, Zaporizhzhia National University, Zaporizhzhia, Ukraine

Corresponding author: Oleksandr Gura Jr. (e-mail: ol.ol.gura@gmail.com).

**ABSTRACT** The article aims to explore the concept of "accessibility" and provide a general context for the creation of an inclusive society in modern conditions, making a special accent on IT industry. As part of the work, an analysis of statistical data from public sources on people with disabilities, as well as legal norms and laws regarding their support and integration into society was carried out among Ukraine, the European Union, and the United States of America. It was found that people with disabilities make up a significant part of modern society, and the legal basis of the developed countries is systematically updated and expanded in the context of supporting people with special needs, placing formal requirements on suppliers of goods and services. On the other hand, it was also found that people with disabilities form a significant group of consumers, so the support of inclusiveness trends has a positive effect on the income of modern companies, from which a conclusion was made about the relevance of an accessibility processes in the conditions of modern business and the IT sphere in particular. During the study, an analysis of current trends in the software development industry in the context of accessibility was carried out, as well as an analysis of web accessibility standards, mainly the "Web Content Accessibility Guidelines (WCAG)" due to its recognition and prevalence, flexibility and adaptability to new technologies, systematic maintenance, and support, as well as integration into the legal framework of the developed countries. Accessibility of the web application was studied in the context of the development lifecycle, at the next stages: requirements analysis, design creation, writing of software code and testing. A model of continuous accessibility process for projects with flexible iterative processes was proposed, which involves a use of technical means, such as static code validators, browser extensions or comprehensive solutions for automated accessibility testing. It was concluded that the comprehensive and systematic use of both organizational and technical accessibility practices at all stages of iterative development is a key factor in the success of the implementation of this initiative in modern software development processes.

**KEYWORDS** web accessibility; software development; software testing; a person with disabilities; inclusivity; automated testing.

## I. INTRODUCTION

ACCORDING to the latest statistics from the World Health Organization, approximately 1 billion of people live with some form of disability, which is about 15% of the world's population. Aspects of disability can vary depending on their nature and degree of severity, covering people of all age and demographic groups. The most common types of disability include:

1. Mobility disorders that affect a person's ability to move freely and independently. Diseases such as paralysis, amputation, cerebral palsy, and multiple types of a sclerosis fall under this category. According to the World Health Organization, about 10% of the world's population, or approximately 650 million people, live with a disability related to mobility.

2. Visual impairments range from partial vision to total blindness. According to the International Agency for the Prevention of Blindness (IAPB), approximately 2.2 billion people worldwide are visually impaired or blind. This includes conditions such as macular degeneration, diabetic retinopathy, and cataracts.

3. Hearing impairment affects the ability to clearly hear sounds and speech. The World Health Organization reports that more than 5% of the world's population, or approximately 430 million people, require rehabilitation to treat hearing loss or deafness.

4. Defects of intellectual development. These disabilities affect cognitive functioning and adaptive behaviour. Conditions such as Down syndrome, autism spectrum disorder, and intellectual disability fall into this category. According to the World Health Organization, approximately 1-3% of the world's population has this type of disability.

5. Psychiatric disorders affect mental health and emotional well-being. Conditions such as depression, anxiety disorders, bipolar disorders, and schizophrenia fall into this category. According to estimates, mental health disorders affect about 10% of the population worldwide.

6. Speech and language disorders affect communication skills, making it difficult to understand language [1-3].

People with disabilities constitute a significant part of society and form a separate social group, which throughout the history of mankind systematically highlighted the relevance of the problem of ensuring inclusiveness in society, which was reflected in the laws of many developed countries. Considering the legislative framework of the United States of America, it is worth noting the following laws that directly relate to people with disabilities:

1. Americans with Disabilities Act (ADA): enacted in 1990, the ADA prohibits discrimination against people with disabilities in all areas of public life, including employment, education, transportation, and public accommodations. It also requires employers to provide reasonable accommodations for qualified individuals with disabilities. Among other regulatory norms, it is worth noting the wide coverage of IT products and infrastructure, especially websites and digital services.

2. Rehabilitation Act: this law prohibits discrimination based on disability in programs and initiatives that receive federal financial assistance. It covers organizations such as schools, colleges, and government agencies. One of its paragraphs, Section 508, requires federal agencies to make their electronic and information technologies accessible to people with disabilities.

For the European Union, in addition to the legislative initiatives of individual countries, the following normative documents are generally accepted:

1. United Nations Convention on the Rights of Persons with Disabilities (UNCRPD): it establishes the rights of persons with disabilities and aims to promote their full participation and inclusion in society. The European Union has ratified the convention and it influences disability policy and legislation in the EU member states.

2. European Accessibility Act: this act, adopted by the European Parliament and the Council in 2019, aims to improve the accessibility of products and services for people with disabilities in the EU internal market. It covers various sectors, including information and communication technology, transport, and banking.

3. Website Accessibility Directive: The Website Accessibility Directive requires government websites and mobile apps to meet certain accessibility standards in the EU member states [4-6, 8, 9].

Examples demonstrating the real legal force of the above-mentioned regulations are, of course, lawsuits against business representatives for non-compliance with the concepts of accessibility of their goods or services. One of the most high-profile cases in recent years is a lawsuit regarding the violation of the rules of web accessibility, namely against the Domino's Pizza company. The accusing party argued that Domino's website and smartphone app were not accessible to people who use screen readers, and thus Domino's digital products violated Title III of the Americans with Disabilities Act (ADA). Specific accessibility barriers cited in the lawsuit included:

- no alt text for graphics;
- empty hyperlinks with no text to identify the purpose of the link;
- redundant links that point to the same URL.

The final court decision stated that the Dominos website and mobile app were not accessible to people with disabilities in accordance with the Americans with Disabilities Act (ADA). Unfortunately, open sources do not provide exact information on the damages suffered by Dominos, but apart from any potential financial costs related to legal fees, a large part of them was undoubtedly spent for updating the digital platforms to meet accessibility standards. In addition, this case significantly affected Domino's reputation and brand perception [7].

The protection of social groups with special needs is also reflected in the legislative framework of Ukraine, namely in the following documents:

1. Law of Ukraine on Social Protection of the Disabled: this law establishes the legal basis for the protection of the rights and provision of social integration of persons with disabilities in Ukraine. It covers various aspects including welfare, employment, education, and accessibility.

2. National program for ensuring equal rights and opportunities for people with disabilities: this program, adopted by the Ukrainian government, outlines measures to promote the integration and participation of people with disabilities in society. It includes initiatives to improve accessibility, health, education, and employment opportunities.

It is worth noting that Ukraine does not have a special national legislation, like the ADA or the European Accessibility Act, regarding the mandatory provision of accessibility in IT services or products. However, Ukraine is a member of the UN and undertakes to support the principles of accessibility and inclusiveness set forth in international conventions and agreements. Although there have been no specific lawsuits or proceedings in the Ukrainian Public Field regarding accessibility violations, certain legal groups and advocacy organizations may sometimes use existing laws and regulations, such as general anti-discrimination laws, to address accessibility violations. In addition, most international companies operating in Ukraine try to adhere to international accessibility standards to ensure that their digital products and services are accessible to foreign users as well [9-11].

It is worth noting that legal regulations and the risks of lawsuits are in no way the main motivator for ensuring the inclusiveness and accessibility of one's processes and products for modern business. As mentioned above, people with special needs are a significant part of modern society, and, accordingly, a separate consumer group. Among the factors of its attractiveness for modern business can be noted:

1. The size of the global market for people with disabilities, which represents a significant consumer segment, estimated of more than $8 trillion annually.

2. Specific shopping patterns. People with disabilities spend money on a wide range of specific goods and services, paying particular attention to products from the health care category, medical and assistive devices, transportation, and other goods that help them normalize their lives. They are also active

consumers in sectors such as technology, fashion, travel, and media content.

4. Impact of inclusivity trends. The very factor of accessibility and support of inclusive trends in products or services often plays a decisive role in the buying decision for people with disabilities. Companies that prioritize accessibility and inclusiveness often gain a competitive advantage and can capture a larger share of the global market.

5. Emphasis on online shopping. With the development of e-commerce, online shopping is becoming more and more popular among people with disabilities. According to a Pew Research Centre survey, 81% of adults with disabilities in the United States have shopped online, compared to only 54% of adults without disabilities.

6. Employment and income. Modern state and private initiatives for the inclusiveness of employment and social benefits have greatly contributed to the level of income of people with disabilities, which, in turn, has had a positive effect on their purchasing power.

The above-mentioned factors did not go unnoticed by modern business, and trends in support of inclusiveness and accessibility were reflected in the policy and development of modern business giants. For example, Apple, known for its commitment to accessibility, has integrated a wide range of features into its products to support users with disabilities. iOS and macOS devices offer features such as Voice Over (a screen reader), Magnifier (to magnify content), Switch Control (for users with limited mobility), and Assistive Touch (to customize touch gestures). These accessibility features enable people with disabilities to use Apple products effectively and independently [12].

### *A. ANALYSIS OF THE LITERATURE AND SOURCES*

Accessibility IT industry today, unfortunately, is not yet a massive popular trend. However, it has attracted more and more attention in recent years and is reflected in the works of leading scientists and industry specialists. The largest number of works, among those reviewed, focused the main attention on the very issue of accessibility in the context of information technologies and reviews of current standards emphasizing the imperfection in the accessibility of key popular resources [13, 17, 18]. Some scientists dived into the subject of the implementation of accessibility concepts in certain technical areas and proposed innovative ways or technologies of accessible development and testing [14-18]. The problem of accessibility of web applications is also a popular subject in the domestic scientific community. Ukrainian scientists have successfully supported the trend of studying technical and organizational aspects of accessibility, such as designing interfaces or testing technologies [19, 20].

Having some attention from the scientific community, accessibility trends in the IT industry are gradually beginning to influence the process of training young specialists in information technologies. The trend towards the implementation of accessibility practices in existing courses or the creation of separate disciplines in the leading educational institutions of the USA:

1. Carnegie Mellon University (CMU). Accessibility is integrated into various courses within the HCI curriculum, with topics such as inclusive design, accessible user interface design and usability testing with people with disabilities. For example, Accessible Computing course (offered as part of the HCI curriculum).

2. University of Washington (UW). The Department of Human Centered Design and Engineering (HCDE) offers courses and research opportunities related to accessibility, including the design and evaluation of inclusive technologies. For example, Design for Accessibility course.

3. Stanford University offers courses and research opportunities related to accessibility in addition to the usual technical disciplines, including computer science and design. For example, Design for Accessibility course [21-23].

A similar trend is also relevant for the leading Ukrainian technical universities:

1. Lviv Polytechnic National University. Accessibility topics can be covered in software development, user interface development, and web development courses. For example, User Interface Design course (offered by the Department of Software Engineering).

2. National Technical University of Ukraine "Ihor Sikorsky Kyiv Polytechnic Institute". Accessibility can be included as part of courses in human-computer interaction, user interface design, and software development methodology. For example, Human-Computer Interaction course (offered by the Faculty of Informatics and Computer Science) [24, 25].

The analysis of scientific sources showed that the context of accessibility and inclusiveness is very popular in the deliverables of modern scientists of various specializations. Relevant practices are studied by them both in the context of design, development, and in the general context of this phenomenon in the modern IT industry. However, in our opinion, the problem of the modern scientific approach to the study of web accessibility lies precisely in its fragmentation and emphasis on individual aspects, while lacking a vision of the general concept of ensuring it as a whole business project process. Despite the detailed and, in many aspects, innovative practices, approaches and tools proposed by modern scientists in the industry, the question of "how to apply them in the conditions of a real IT project?", in our opinion, remains unanswered.

This thesis is confirmed by the above analysis of training programs for IT specialists in flagship technical universities - despite the inclusion of topics related to accessibility in the programs of various disciplines, it is still not aggregated into a separate course, as well as is not considered by students and teachers in a general context of the software development process and project management issues, but simply serves as an optional supplement to technical disciplines.

The purpose of the research is to analyze existing standards and approaches to accessibility in the context of their complex application in the real-world IT projects, to ensure a continuous and controlled process.

## II. THEORETICAL ASPECTS OF THE RESEARCH

Accessibility in the context of web development is achieved through design and coding practices that ensure digital content is accessible, comprehensible, and reliable for all customers, regardless of their abilities or physical limitations. Among them, it is worth noting the key ones:

1. Semantic markup. Semantic HTML markup provides a framework for accessibility by structuring web content in a meaningful way. The use of appropriate HTML elements, such as headings, paragraphs, lists, and semantic guides (such as

navigation, header, and footer), allows specialized technical devices used by people with disabilities to understand and interpret content correctly.

2. Accessibility of the keyboard. Websites and web applications should be able to be navigated and controlled using only the keyboard, without the use of a mouse or other pointing device. Keyboard accessibility ensures that users with limited mobility or those who cannot use a mouse can interact with all interactive elements and easily navigate the content of the resource.

3. Alternative text for images. Providing descriptive alt text for images is critical for users who cannot see visual content, such as the visually impaired or users who have disabled images in their browsers.

4. Color contrast and visual design. Providing sufficient color contrast between text and background elements improves readability and usability for users with low vision or color blindness. Additionally, thoughtful visual design, such as clear typography and adequate padding, contributes to the overall accessibility and quality of the app's user experience.

5. Accessible forms and controls. Forms and interactive elements must be designed and implemented in such a way that they are accessible to all users. This includes providing labels for form fields, ensuring proper focus control, and using accessible markup for interactive controls such as buttons and links.

Considering a technical complexity of even the simplest modern web applications, the fulfilment of accessibility requirements can only be possible if there is a comprehensive and generally accepted standard that will unambiguously and systematically declare clear technical requirements regarding the specific accessibility norms.

The Web Content Accessibility Guidelines (WCAG) is a set of technical standards developed by the World Wide Web Consortium (W3C) to make web content accessible to people with disabilities. WCAG provides a comprehensive framework for creating accessible websites and web applications, covering a wide range of limitations and violations. Development of WCAG began in the late 1990s as a response to the growing need for accessibility standards in web development. The first version, WCAG 1.0, was published in 1999, followed by WCAG 2.0 in 2008, and the latest version, WCAG 2.1, was published in 2018. The Web Content Accessibility Guidelines (WCAG) are considered the primary and most widely used accessibility standard worldwide for the following reasons:

1. International recognition: WCAG is developed and maintained by the World Wide Web Consortium (W3C), an international community that is the authoritative and recognized source of standards for a range of current web practices and initiatives by most of the giants of the web industry.

2. Full coverage. WCAG targets a wide range of disabilities, including visual, hearing, physical, language, cognitive, learning and neurological.

3. Development and maintenance. WCAG is systematically supplemented and expanded in parallel with the progress of technology and changes in user needs. Since its initial release in 1999, it has undergone several revisions, with the latest version WCAG 2.1 published in 2018.

4. Synchronization with laws and policies. WCAG is frequently referenced in laws, regulations, and implementations related to web accessibility worldwide. Many countries and organizations adopt WCAG as the basis of their accessibility requirements, making it the de facto standard for legal compliance.

5. Community consensus. WCAG was developed through a consensus process involving stakeholders from industry, government, scientists, and advocacy groups. This collaborative approach ensures that WCAG reflects the diverse perspectives and experiences of the accessibility community.

6. Practical instructions. WCAG provides practical guidelines and methods for implementing accessibility features and addressing common accessibility barriers. It includes examples, explanations and resources to help web developers, designers, content writers and policy makers understand and effectively apply the guidelines [26].

### A. ACCESSIBILITY IN THE CONTEXT OF WEB APPLICATION DEVELOPMENT PHASES

To study the conditions of implementation of accessibility practices in the process of web application development, it is crucial to understand the lifecycle of software development itself, its main stages and phases. Although specific methodologies and approaches to development may vary, the following are the main stages that are relevant to any type of web application, regardless of its domain and technical characteristics:

1. Planning and collection of requirements. The main objective of the stage is to determine the scope of the project, its objectives and key requirements. The result of successful completion of this phase is usually a finalized list of basic requirements, a basic layout and a project plan.

2. Design. The main objective of the stage is to create a visual and structural design of the web application. Output should include frames, prototypes, user interface designs, styles.

3. Development. The aim of this stage is directly to create a website based on the approved design and requirements. It includes writing code, developing external and internal functions, integrating third-party services or APIs, implementing adaptive design for different devices, performing modular testing.

4. Testing, as one of the stages of the development process, aims to ensure that the website meets the standards of both previously formulated requirements and generally adopted ones, including accessibility. It consists of performing different types of testing, such as functional testing, user-friendliness testing, compatibility testing (cross-browser, cross-device), performance, accessibility, security.

Depending on the chosen development paradigm, these phases can be repeated or shuffled, which means that feedback and adjustments can take place throughout the development process. For example, one of the most common web application development methodologies in the modern industry is SCRUM, the basis of which is the division of development into short iterations, each of which includes all the above stages.

### B. ANALYSIS OF REQUIREMENTS

The stage of requirements analysis is usually the most responsible phase of the entire development process because it builds an entire vision of its functional and non-functional features. This statement is relevant in the context of ensuring accessibility, the priority of which must be established at the level of formulation of requirements and choosing the

implementation mechanisms.

In the requirements verification phase of web application development, to ensure compliance with accessibility standards, it is important to focus on several key factors:

1. Accessibility requirements: accessibility requirements are clearly defined and documented in the project specifications, while the strategic decision to develop an accessible product that meets the standards is taken.

2. User scenarios analysis from the perspective of people with disabilities: functionalities and design requirements of the app should be based on the needs of all users, including those with a disability. It is therefore important to ensure that scenarios that include user-facing and interaction functions are analyzed and, from this perspective, properly addressed and highlighted in the whole set of requirements.

3. Comprehensive functional specifications should contain functions of accessibility and interaction. The availability of keyboard accessibility, compatibility with a screen reader, alternative text for images and variable text size for people with disabilities are essential.

### C. DESIGN

The web application and layout design accessibility take place in different aspects and special requirements to the user interface to make sure that it is accessible to all users, including people with disabilities. The main aspects of design accessibility specified in the WCAG standard include:

1. Color Contrast: provision of sufficient color contrast between the text and the background elements to improve readability for users with weak vision or long-sightedness. Layouts must use colors that meet accessibility standards and guidelines, such as the Web Content Accessibility Policy.

2. Interactive elements such as buttons, links and form fields must be properly selected and focused. Layouts should contain clear and descriptive labels for interactive elements to help users who rely on screen readers or keyboard typers.

3. Focus indicators: to provide visible focus indicators for interactive elements that indicate their current state or activity. The patterns should include visual hints such as contours or highlights to indicate when an item changes its state. This helps customers using keyboard-only input to understand which object on the page they are interacting with at the moment.

4. Navigation: to evaluate the navigation structure and layout of the web application and provide intuitive and user-friendly navigation for all users. Layouts should contain clear navigation menus, navigation paths and landmarks to help users navigate and navigate content effectively. For example, the basic practice of placing the close button in the top right corner of an item will allow users to close forms on a page intuitively and without having to search for a long corresponding character.

5. Alternate Text: to include descriptive alternative text (alternate text) for images and non-text content to provide context and meaning for users who do not see visual content. The layouts should specify alternative text for images and multimedia elements to make sure that they are available to users of the screen reader.

6. Adaptive design: to ensure that the web application structure adapts to different screen sizes and devices. The layouts should consider how the frame of the app and the content will be rearranged and adjusted depending on the size of the view window, ensuring coherent and accessible user interaction on all devices. This aspect is extremely relevant in the context of accessibility, because most specialized devices and tools for people with special needs have non-standard screen proportions or additional windows/elements in the browser, affecting the effective size and format of the display area of the site.

### D. DEVELOPMENT

Describing the “continuous” and multi-phase approach to ensuring accessibility, let us assume that the designs and layouts of the pages are already provided with accessibility, while the requirements for both the technical and functional aspects are already covering the standards of accessibility. Thus, at the stage of programming/coding itself, the main emphasis should be focused on fulfilling technical accessibility requirements. It is assumed that the design of the page element, its text and animations are already accessible, and the user experience in the context of using this element has also passed through approval by relevant accessibility experts. So, the task of the developer is to focus on the direct implementation, namely the correct use of the internal configurations and states of the element, as well as its placement and parameterization in the correct structural parts of the code of the page.

Working with the code of the web page in modern reality of development in any case goes through such phases as writing and compilation/deployment. The result of the first – the code itself written by the developer in specialized tools. It represents a set of text files and is not yet carrying any business value. The result of the second – already built and functional web page, which can be perceived by the user as a finished product. The transition from phase one to phase two is a process that usually requires additional actions, use of computing resources, external dependencies, and integration with other systems. For example, the code of the "Pay now" button on some commercial web site can be readable at any time by just opening a proper file, while the ability to see this button on a webpage in a browser may require the development of other dependent elements, the presence of translations of its text into different languages in the database, the server infrastructure that processes the button-clicking event and other factors. Accordingly, to optimize the process, it is appropriate to consider ensuring the accessibility in the programming process from two aspects in parallel – accessibility of the raw code base separately, and the already built and deployed application as a whole. For this purpose, two types of tools can be identified globally: code linters and tools for automated verification of compiled pages.

“Linter” is a tool for ongoing verification of raw code during its development. It is not able to carry out a full analysis of the target application, however, it can find semantic or structural errors thanks to static analysis. Returning to the example of the “Pay now” button, given in previous section, the task of the linter is to indicate that the code

```
    <div class = ‘bigRedButton’ id=’payNowButton’>
Pay Now </div>
```

does not meet the accessibility standard, because it does not have “role” attribute inside the “div” tag, which indicates that the exact element is a button, and, therefore, the principle “Acceptability” of the WCAG standard is violated. The correct implementation may be the following:

```
    <div tabindex="0" role="button" aria-
pressed="false" class = 'bigRedButton'
id='payNowButton'> Pay Now </div>
```

An example of a tool that implements the above features is ESLint Accessibility, a plugin for ESlint, a popular static code analysis tool that helps developers identify and solve accessibility issues. It integrates into existing workflows, allowing developers to perform automatic accessibility checks as part of the coding process [27].

The main functions of the ESLint accessibility tool include:

1. Automatic Accessibility Testing: the plugin automatically scans the JavaScript code for features issues, including breaches of general accessibility standards, practices and guidelines. It identifies typical problems with special features related to DOM manipulation, event processing, keyboard navigation, and so on.

2. Customized validation rules: ESLint accessibility comes with a set of predefined accessibility rules based on industry standards, mainly WCAG. Developers can customize the set of rules according to specific project requirements, level of compliance and specific technologies or coding conditions.

3. Continuous integration support: ESLint is compatible with Continuous Integration (CI) systems such as Jenkins, Travis CI and GitHub Actions. Developers can automate accessibility checking as part of their CI pipelines, ensuring consistent accessibility testing in development and deployment environments.

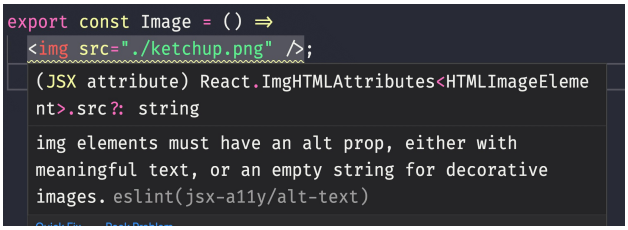


Figure 1. ESLint Accessibility: automatic comment on the lack of alternative text for the image

In general, the ESLint accessibility checking tool helps developers to create more accessible code, identifying and solving accessibility problems in the early stages of the development process. It is worth noting that the above tool is not a guarantee of full accessibility standards compliance, because it does not interact directly with the finished page. Tools for automated page checks is a separate class of programs, the main purpose of which is both visual and structural analysis of the page directly in the browser and search for accessibility defects on it.

A representative example of this type of tools is the AXE DevTools browser extension, designed to help developers identify and solve accessibility problems in web applications. Developed by Deque Systems, the AXE DevTools easily integrates into popular web browsers such as Google Chrome and Mozilla Firefox, providing developers with real-time feedback and guidance on accessibility best practices. The main functions of the AXE DevTools extension include:

1. Automatic Accessibility Testing: AXE DevTools automatically scans web pages for problems, including violations of accessibility standards, such as WCAG. It identifies default features related to HTML tagging, ARIA roles, keyboard navigation, and more.

2. Detailed Accessibility Reports: the extension generates detailed accessibility reports that highlight the problems identified and provides effective advice on how to solve them. Developers can review reports to understand the nature of each accessibility violation and prioritize fixes based on severity and impact.

3. Dev tools integration: AXE DevTools easily integrates with browser developer tools and IDEs, allowing engineers to access accessibility testing functions directly within an existing workflow. This makes it easy to detect and fix problems while coding and configuring web applications [28].

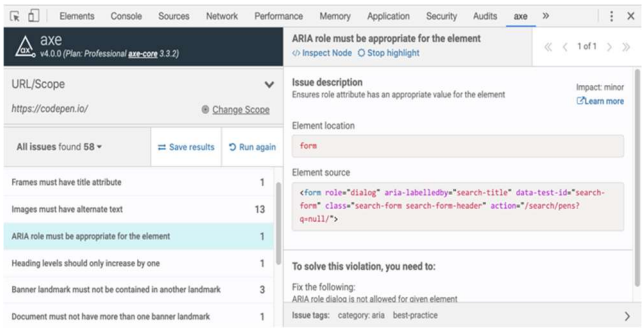


Figure 2 AxeDevTools testing results example

Overall, the AXE DevTools browser extension provides developers with a convenient and efficient way to perform automated accessibility testing by identifying and addressing accessibility issues in the early stages of the development process.

### *E. TESTING*

The testing phase is undoubtedly the key stage in the software development cycle, the main purpose of which is to ensure the conformity of the product to the requirements and standards set by the customer. In the realities of modern projects testing can be carried out in two forms – manual and automated. Manual testing involves professionals who interact directly with the application or website being tested. Testers can explore application functionality, user interface and real-time usability, allowing flexible and creative approaches. At the same time, automated testing involves the use of specialized tools or scripts to perform tests and verify the behavior of the program without a direct involvement of an engineer. It is worth noting that in modern design practices automation is perceived as an initiative supplementary to manual testing and almost never acts as a substitute for the latter: the initial verification of product conformity is always performed manually, and then, with the relevance of this verification for regression testing, the technical possibility and the corresponding concept adopted on the project, an automated test is written.

Automated testing is extremely relevant in the context of accessibility, which is expressed in the following factors:

1. Emphasis on HTML testing: because most of the criteria in the accessibility standards relate primarily to the HTML tag, their testing is algorithmic and objective, which fits perfectly into the concept of automated testing.

2. A large number of checks due to the interdependence of the codebase elements: because the HTML elements of the page are related to each other both syntactically and

semantically (headers go in a clear sequence, and the elements on the page are located in each other), the presence of a defect in one element can provoke a defect in another, which leads to the need to test each of them separately, as well as in different combinations and functional states (for example, if the "Pay" button is present on all components with different methods of payment, the correct practice will be to test the accessibility of the button itself, separately each component, and each pair of "component + button"). A large number of identical repeated checks is the main candidate for automation.

3. Compatibility of accessibility testing with functional automated testing – since the accessibility concept itself is to give people with disabilities a way to use the functionality of the site, accessibility testing may have the same scenarios, data and flows as the functional one. Accordingly, automated accessibility tests typically require the same infrastructure as functional tests, making an automation of accessibility checks, in addition to functional ones, a profitable investment of time. Let us assume that the project infrastructure already has an automated test that registers the user, adds the product to the cart and goes to the payment page, checking that the "Pay now" button is present and active. In this case, to automatically verify the HTML code of this button for its compliance with accessibility standards, the tester will simply reuse the already available code snippets of registration and promotion of the user to the payment page.

An example of one of the popular and relevant tools of automation of functional testing, which includes tools for accessibility testing, is Playwright JS. Playwright is a JavaScript library designed to automate web browser interactions and simulate user actions. The tool, developed by Microsoft, provides a high-level API to perform actions such as web page navigation, interaction with elements, and page content evaluation. It offers a wide set of features such as multi-threaded automation, headless mode for running tests without a visible browser window, and support for testing sophisticated scenarios, including authentication and file uploading. In addition to the generally accepted practices for automation of functional testing, the tool has a block of accessibility checking functions:

1. Integration with Axe: Playwright's accessibility checks use Deque Axe's library, which is the author of many other accessibility tools, including those discussed above. The uniformity and predictability of toolset allows more specifically synchronize the process of their use at different phases.

2. Flexible accessibility configuration: Playwright allows testers to configure which accessibility checks to perform based on their specific requirements and priorities for each development phase separately.

3. Generating Accessibility Reports: Playwright creates detailed accessibility reports that highlight the problems identified and provide effective advice on how to solve them. Developers can review the reports to understand the nature of each accessibility violation and prioritize fixes accordingly.

Playwright's accessibility checking features provide testers with a powerful tool to automate accessibility testing and ensure that web applications are accessible to all users [29].

## III. RESULTS

For the recent year agile methodologies have become the dominant approach in modern IT project management. According to the 14th Annual Agile Status Report by VersionOne, which surveyed more than 1,300 respondents in different industries worldwide, 97% of organizations surveyed stated that they practice Agile development methods. In addition, other industry surveys and reports from reputable sources, such as Gartner, Forrester and Agile Alliance, have also highlighted the widespread implementation of Agile methodologies in IT projects. While the exact percentage may vary slightly depending on the source and the specific criteria used to assess, Agile has become the main approach to managing IT projects. The flexible methodology in IT emphasizes iterative development, collaboration, flexibility, and continuous customer response. Its essence is to break down projects into smaller, controlled phases called iterations or sprints, where inter-functional teams work closely together to provide a working version of the final product as a result of the end of each iteration.

Depending on the specifics of each individual project and the selected flexible methodology, the composition and individual phases of iterations may vary, however, conceptually each iteration usually consists of:

1. Design and planning, the main purpose of which is the preparation of designs and requirements for further development, as well as the planning of work for the next repetition, considering the capabilities of commands, the complexity of tasks and the priority of requirements.

2. Development, during which the direct writing of the software code, compilation and testing of the project is carried out. Testing and release during which iteration ends and the results are transmitted to the customer for acceptance of the work performed.

Because the individual accessibility tools and practices discussed in previous sections relate to the respective phases of each iteration, the resulting accessibility process is appropriate to be considered precisely in the context of iterations in the flexible development methodology.

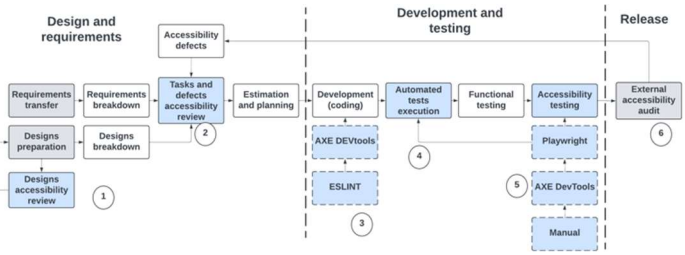


Figure 3. Model of continuous accessibility assurance process

Figure 3 shows the accessibility process adopted on a real commercial web project to develop an online store. It is worth noting that the project development team, in addition to the classic positions of developers, testers, managers and customer representatives, also contains an accessibility expert responsible for the internal audit and support of the process at all its internal stages. Let us review the main phases of the process (Table 1).

**Table 1. Phases of the continuous accessibility process in iterative development methodology**

| Phase | Responsible team members | Schedule | Input data | Goal and value |
|---|---|---|---|---|
| 1 Design review | Accessibility expert<br>Designers<br>Business analysts | Before the start of the development phase of each iteration, during the transfer of designs and requirements | Interface design in the format of pictures and/or interactive layout | Determine whether the proposed user interface designs meet the accessibility requirements. Give recommendations on how to solve the problems identified, agree changes with the customer and pass the result to the development team |
| 2 Functional requirements review and estimation | Accessibility expert<br>Business analysts | Before the start of the development phase of each iteration, during the planning of tasks for the next iteration | List of defects found during the testing of the previous iteration. List of tasks considered to be taken in the current iteration | Analyze the list of the tasks and defects to be fixed, scheduled to be performed in the next iteration, for violation of accessibility requirements. Make appropriate adjustments to the content of requirements and tasks, add recommendations to their implementation in order to ensure accessibility. Prioritize the accessibility defects found in previous iterations, provide recommendations on their effective elimination |
| 3 Accessible development | Developers | According to the results of planning | Designs and requirements that meet accessibility standards | Conduct the development of the planned functionality, taking into account common international accessibility standards, accessibility requirements introduced by the expert at the previous step, as well as using accessibility tools such as linters and automatic page testing programs (e.g., ESLint and AxeDevTools) |
| 4 Automated accessibility validation | Developers<br>Testers | According to the estimation results, after the end of the code creation. | Code base that was written or modified during the task. Automatic accessibility tests written by the testing team in previous iterations | Perform automatic regression testing of the project's code base using a set of automatic tests written by the test team in previous project iterations. This determines whether the code changes associated with the execution of the current task have affected the compliance with the accessibility criteria of the functionality developed in previous iterations. In the case if tests show a negative result, the analysis of test reports is carried out, and the task is sent for fixing |
| 5 Accessibility testing. Writing automated accessibility tests | Testers | According to the results of planning, after the end of the development of the task | Codebase that has been automatically checked by regression accessibility tests | Conduct a comprehensive accessibility testing, which includes the following steps:<br>1. Check whether the functionality meets the basic accessibility standards implemented by WCAG<br>2. Check whether the functionality meets the requirements and recommendations implemented by the expert.<br>3. Test functionality with automated tools (AxeDevTools) as well as specialized software for people with disabilities.<br>4. Create automated regression scenarios for testing of this feature during subsequent iterations.<br>5. Document the defects found during the accessibility testing. Pass them for analysis to the expert |
| 6 External accessibility audits | Customer representative/external experts | After the transfer of the work done by the development team | A ready-made web application available for use in an internal or external network infrastructure | Conduct acceptance testing of the functionality with the help of business representatives or external auditors to verify the quality of performed work and pass the found defects to the development team for fixing in the following iterations. |

## IV. CONCLUSIONS

The process discussed in the previous chapter is not universal and depends on a few factors, such as:

- staffing of the development team and the customer team, the availability of an accessibility expert and a team of external auditors, the presence of accessibility testing qualifications in the test team;
- selected methodology of development, stages of the cycle of iterations;
- availability of technical qualifications and legal ability to use tools of automated accessibility testing;
- development timetables and budgets, accessibility requirements priority and customer readiness to increase development costs for them.

However, the overall concept of introducing exclusive practices to ensure accessibility at each of the phases of iterative software development gives the following positive effects:

1. Controllability and transparency of the process at each stage: both the development team and the customer have an opportunity to obtain comprehensive information about the state of the product in the context of accessibility in any phase of development, as well as to make adjustments in the process

of its provision, priority and desired result at each phase. For example, the systematic presence of a large number of defects detected during the transfer of the design to the development team can be a reason for introducing additional practices, such as conducting specialized training for the design team, or the introduction of automation tools. At the same time, the other stages of the process will remain unchanged.

2. "Shift-left" and cost reduction of defects: each level of accessibility is tested as early as possible, minimizing the time and resources needed to correct potential defects and eliminate dependencies. For example, if the contrast design of the "Pay" button, passed to the development team for implementation, does not meet the accessibility criteria, this fact will be known to all the team members even before the start of code development. Accordingly, the developers will manage to redistribute the task to the next iteration, the designers team – to eliminate the defect, while the time of the developers will not be wasted on the implementation of the deliberately defective design, and capacity of functional testers – on the detection of a defect.

3. Scalability and flexibility of the process: in case of changes in team composition, technologies, external conditions or development priorities, it is sufficient to make changes to the relevant phases of the accessibility process, while still saving its overall concept and, accordingly, control over the result. As an example, let us assume that there are unexpected technical problems in the project infrastructure that make it impossible to perform automated accessibility verification by linters during the next iteration, which may lead to increased manual testing time and the risks of detection of defects at a later stage. However, this potential situation does not put at risk the final accessibility of the product and will not affect the final experience of the end consumer, as there are still several phases of testing planned later – the team loses a little time but retains the product quality and its own reputation.

In our opinion, the comprehensive and systematic use of both organizational and technical accessibility practices at all stages of iterative software development is a key factor in the success of the implementation of this initiative in modern software development processes, which allows the customer and project management to be confident in compliance with the laws and worldwide standards, and users – to receive a stable quality and affordable service regardless of the context of their own health.

## References


[1] J. van der Geer, J. A. J. Hanraads, R. A. Lupton, "The art of writing a scientific article," *Journal of Science Communication*, vol. 16, issue 3, pp. 51-59, 2000.

[2] World Health Organization, "Global report on health equity for persons with disabilities: executive summary," Geneva, 2022, [Online]. Available at: https://www.who.int/publications/i/item/9789240063624

[3] C. S. Brennan, "Disability rights during the pandemic: A global report on findings of the COVID-19 disability rights monitor," COVID-19 Disability Rights Monitor, 2020, [Online]. Available at: https://www.internationaldisabilityalliance.org/sites/default/files/disability_rights_during_the_pandemic_report_web_pdf_1.pdf

[4] E. J. Williamson et al., "Risks of COVID-19 hospital admission and death for people with learning disability: population-based cohort study using the OpenSAFELY platform," BMJ, vol. 374, p. n1592, 2021. https://doi.org/10.1136/bmj.n1592.

[5] U.S. Census Bureau, "Health insurance coverage in the United States: 2019," [Online]. Available at: https://www.census.gov/data/tables/2019/demo/health-insurance/health-insurance-coverage.html

[6] European Commission, Eurostat, "People with disabilities statistics," [Online]. Available at: https://ec.europa.eu/eurostat/statistics-explained/index.php?title=People_with_disabilities_statistics.

[7] World Bank, "Disability", [Online]. Available at: https://www.worldbank.org/en/topic/disability/overview.

[8] U.S. Department of Justice, "Justice department settles with Domino's Pizza to ensure accessibility for people with disabilities," [Online]. Available at: https://www.ada.gov/dominos_hoa/press_release.html.

[9] U.S. Department of Justice, "Introduction to the ADA," [Online]. Available at: https://www.ada.gov/ada_intro.html.

[10] European Commission, "Accessibility", [Online]. Available at: https://ec.europa.eu/social/main.jsp?catId=1202&langId=en&intPageId=1436.

[11] Verkhovna Rada of Ukraine, "Law of Ukraine on the fundamentals of social protection of the disabled in Ukraine," [Online]. Available at: https://zakon.rada.gov.ua/laws/show/875-12. (in Ukrainian)

[12] Cabinet of Ministers of Ukraine, "On amendments to the national program for ensuring equal rights and opportunities for persons with disabilities for 2011-2015," [Online]. Available at: https://www.kmu.gov.ua/npa-pro-vnesennya-zmin-do-nacionalnoyi-programi-zabezpechennya-rovnyh-prav-ta-mozhlivostej-dlya-osib-z-invalidnistyu-na-2011-2015-roki (in Ukrainian)

[13] Apple Inc., "Accessibility," [Online]. Available at: https://www.apple.com/accessibility/.

[14] F. D. Arini and F. Fitri, "How accessible the university websites in Indonesia for people with disabilities," *IJDS Indonesian Journal of Disability Studies*, vol. 7, pp. 164-169, 2020. https://doi.org/10.21776/ub.ijds.2020.007.02.05.

[15] A. Nuñez, A. Moquillaza, and F. Paz, "Web accessibility evaluation methods: A systematic review," *Advances in Human Factors and Systems Interaction*, vol. 965, pp. 238-247, Springer, 2019. https://doi.org/10.1007/978-3-030-23535-2_17.

[16] S. Henry, "Understanding web accessibility," In Foundations of Human-Computer Interaction, pp. 1-15, Springer, 2006. https://doi.org/10.1007/978-1-4302-0188-5_1.

[17] J. Zhang, P. Purgathofer, M. Brereton, G. Fitzpatrick, and F. Güldenpfennig, "Handle the way: Enhancing Web accessibility for people with disability," pp. 117-120, 2016. https://doi.org/10.1145/2908805.2909403.

[18] I. Basdekis, I. Klironomos, I. Metaxas, and C. Stephanidis, "An overview of web accessibility in Greece: A comparative study 2004-2008," *Universal Access in the Information Society*, vol. 9, pp. 185-190, 2010. https://doi.org/10.1007/s10209-009-0166-z.

[19] S. Maatta, "Web accessibility assessment of urban public library websites," *Public Library Quarterly*, vol. 33, pp. 187-204, 2014. https://doi.org/10.1080/01616846.2014.937207.

[20] V. Klym, Y. Tarasenko, "Testing of Web resources for inclusivity: Audit for reengineering," Bulletin of Vinnytsia Polytechnic Institute, no. 3, pp. 60-66, 2022. https://doi.org/10.31649/1997-9266-2022-162-3-60-66. (in Ukrainian)

[21] L. Krestyanpol, M. Kovalchuk, O. Vashkevych, "Web accessibility as the main means of breaking down digital barriers for people with disabilities," *Applied Problems of Computer Science, Security and Mathematics*, vol. 1, May 2023, pp. 4-9.

[22] Carnegie Mellon University - Human-Computer Interaction Institute, [Online]. Available at: https://hcii.cmu.edu/academics.

[23] University of Washington - Human Centered Design & Engineering Department, [Online]. Available at: https://www.hcde.washington.edu/.

[24] Stanford University - Graduate School of Education, [Online]. Available at: https://ed.stanford.edu/.

[25] Lviv Polytechnic National University - Department of Software Engineering, [Online]. Available at: https://lpnu.ua.

[26] National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute" - Faculty of Informatics and Computer Science, [Online]. Available at: https://fiot.kpi.ua.

[27] W3C - Web Content Accessibility Guidelines (WCAG) Recommendations, [Online]. Available at: https://www.w3.org/WAI/standards-guidelines/wcag/.

[28] Eslint accessibility plugin, [Online]. Available at: https://www.npmjs.com/package/eslint-plugin-jsx-a11y.

[29] AXE Dev tools, [Online]. Available at: https://www.deque.com/axe/devtools/.

[30] Playwright accessibility, [Online]. Available at: https://playwright.dev/docs/accessibility-testing.

***OLEKSANDR GURA.,*** *Doctor of Pedagogical Sciences, Professor of the Department of Psychology, Zaporizhzhia National University, Zaporizhzhia, Ukraine.*
*ORCID ID 0000-0001-6720-2481*
*guru.olexandr@gmail.com*

***OLEKSANDR GURA JR.**, Ph. D., doctorate student, Zaporizhzhia National University, Zaporizhzhia, Ukraine.*
*ORCID ID 0009-0000-2283-3941*
*ol.ol.gura*@gmail.com