

## METHODS FOR IMPLEMENTING ACCESSIBILITY IN USER INTERFACE DESIGN

L. Bovnegra, S. Kutsyn

National Odessa Polytechnic University  
1, Shevchenko Ave, Odesa, 65044, Ukraine  
Emails: dlv5@ukr.net, 7813boychenko@gmail.com

Accessibility has become a first-class quality attribute in UI/UX and web design, affecting legal compliance, user equity, and business outcomes. However, teams often face practical barriers when translating high-level guidelines into day-to-day workflows and measurable engineering outcomes. This article consolidates established methods for implementing accessibility in user interface design and presents a reproducible workflow that integrates standards-driven evaluation, incremental code refactoring, and user testing with assistive technologies. The study contributes to computer science and HCI by bridging normative guidance with process-level tactics for UX and front-end teams, yielding actionable artefacts (metrics, checklists, mapping tables) suitable for continuous delivery environments. We synthesize literature and standards to define a stepwise pipeline: heuristic and WCAG-oriented audits; prioritization of barriers by severity and user impact; minimal-change refactoring (semantic HTML, ARIA roles/states, focus management, keyboard navigation, contrast, error handling); validation via manual checks and automated tooling, plus user testing with screen readers; regression control in CI/CD. The evaluation model includes Accessibility Conformance Ratio, accessibility defect density, task success rate, time-on-task, error rate, and an effort-normalized efficiency metric. Applied to a representative web module, the workflow led to a systematic increase in conformance coverage and a reduction in critical barriers, accompanied by higher task success and shorter completion times for users relying on assistive technologies. Statistical tests on pre/post measures support the significance of the observed improvements, while threats to validity and transferability are discussed. The paper offers a reusable accessibility engineering approach, a compact measurement model, and traceability from WCAG criteria to UI/code changes and tests. The results inform UI/UX designers and developers how to stage accessibility improvements without disruptive redesigns, and how to embed accessibility gates into continuous delivery.

**Keywords:** accessibility; WCAG; UI/UX; web design; user testing; WAI-ARIA; refactoring.

**Introduction.** Accessibility of digital interfaces today is not only an ethical and social requirement but also a clearly defined engineering task that directly influences audience reach, user experience (UX) quality, and business results. According to WHO estimates, 1.3 billion people - about one in six globally ( $\approx 16\%$ ) - live with significant forms of disability; therefore, every UI/UX and web-design project inherently interacts with a large and diverse group of users with specific interface and interaction needs (AT support, keyboard navigation, contrast, etc.). Comparable estimates for individual countries (e.g., 28.7% of adults in the United States) further underscore the scale of the phenomenon and the need for a systematic engineering response.

At the EU regulatory level, the European Accessibility Act (Directive (EU) 2019/882) takes effect from 28 June 2025, extending requirements to a broad range of private-sector products and services (including e-commerce, e-books, banking services, mobile apps, self-service terminals). Previously, Directive (EU) 2016/2102 already applied to the accessibility of websites and mobile applications of public-sector bodies. Both initiatives are based on the harmonized standard EN 301 549 (version 3.2.1), which in turn relies on WCAG 2.1 criteria, ensuring consistency of technical requirements and assessment mechanisms. For developers outside the EU, the U.S. Section 508 (Revised 2017) is likewise illustrative, incorporating WCAG 2.0 AA as the normative basis for the public sector. Taken together, this forms a global

landscape in which accessibility becomes a matter not only of voluntary practice but also of formal obligations and audits [1].

Despite the maturity of standards, the actual state of web accessibility remains problematic. According to WebAIM Million 2025, 94.8% of homepages of leading sites had detected WCAG 2 violations (down from 95.9% in 2024), with an average of  $\approx 51$  errors per page (50,960,288 errors across 1 million pages). The most common issues are low contrast (79.1% of pages), missing image alt text (55.5%), missing form labels (48.2%), “empty” links (45.4%), etc.; 96% of all detected errors fall within just six categories. At the same time, interface structural complexity is increasing: the average number of elements on a homepage rose to 1,257 (+7.1% year-over-year), and ARIA attribute usage to  $\approx 106$  per page (+18.5% y/y), correlating with an increase in errors. These data document a systemic gap between normative requirements and current UI/UX development practices.

The cost of non-inclusivity is measured not only in missed interaction opportunities but also in direct economic impact. According to the Click-Away Pound 2019 survey, 69% of internet users with disabilities abandon websites when they encounter barriers, and lost online sales in the United Kingdom reached £17.1 billion per year. Moreover, 83% of respondents limit their purchases to websites whose accessibility they trust. These facts strengthen the business case for systematic accessibility work as part of customer acquisition and retention strategies [1].

WCAG 2.2 standards (W3C Recommendation as of October 5, 2023) clarified expectations for several critical interaction scenarios—adding nine new success criteria (including 2.4.11 Focus Appearance, 2.5.7 Dragging Movements, 2.5.8 Target Size) and refining approaches to internationalization. This expansion is intended to improve real-world usage, especially on mobile devices and within complex web applications. Together with EN 301 549 and national regulations, it provides a clear axis for measuring conformance and planning engineering changes.

Despite existing guidance, the industry often faces three practical challenges.

First, the over-saturation of interfaces with third-party components (ad networks, widgets, CMP banners) complicates accessibility control: studies indicate that advertising can increase WCAG violation rates across a large share of sites due to incorrect focus management, labeling, and element behavior.

Second, cookie notices and consent banners often exhibit contrast, semantic, and screen-reader/announcability defects for users with visual impairments, creating additional entry barriers.

Third, design dark patterns can manipulate choices around consent and privacy, undermining the WCAG principles of understandability and operability [2].

In this context, an approach is needed that combines normative clarity (WCAG/WAI-ARIA, EN 301 549, Section 508) with a realistic process for engineering change in the product. Industry experience shows that incremental accessibility adoption (via regular audits, impact-based barrier prioritization, minimal code changes followed by validation, and regression control in CI/CD) enables tangible reductions in defects and increases in conformance without “big-bang” rewrites or development freezes. From a computer-science perspective, this requires formalizing metrics (e.g., Accessibility Conformance Ratio, defect density, time-on-task and success rate in user testing with AT) and models for estimating the effect of changes to justify architecture, design, and backlog prioritization decisions. In the following sections, we align the accessibility implementation process with normative criteria and demonstrate how stepwise refactoring raises WCAG 2.2 conformance and improves UX indicators in a representative web case study [2].

**Objective.** To systematize literature-validated methods for implementing accessibility in user interface (UI) design and to present a reproducible engineering process that combines standards-oriented audits (WCAG/WAI-ARIA), incremental refactoring, and user testing with

assistive technologies (AT). This process should lead to measurable increases in conformance and improvements in UX metrics in a web case study.

Research Questions (RQ):

- RQ1. To what extent do accessibility-barrier density decreases and the Accessibility Conformance Ratio (ACR) increase after one or multiple incremental cycles?
- RQ2. Do user-testing results with AT (task success, time-on-task, error frequency) improve after targeted changes in semantics, focus, contrast, navigation, etc.?
- RQ3. What is the relative cost-effectiveness of refactoring (effect per unit of engineering time), and how does it compare to alternatives (e.g., a “redesign from scratch”)?

Context and relevance. Approximately 1.3 billion people worldwide (~16%) live with significant forms of disability; in the U.S., about 28.7% of the adult population. According to WebAIM Million 2025, 94.8% of homepages have automatically detected WCAG 2 deviations, with an average of about 51 errors per page. At the same time, regulation is tightening (the European Accessibility Act effective June 28, 2025; harmonization with EN 301 549; in the U.S., Section 508) [1].

Scientific significance. Positioned at the intersection of computer science, HCI, and software engineering, this work bridges the gap between normative standards (WCAG 2.1/2.2, WAI-ARIA) and process tactics for UI/UX teams and front-end developers. The paper’s contributions include: (i) a reproducible implementation pipeline; (ii) a measurement model (ACR, defect density, user-test indicators, cost-efficiency E); (iii) traceability from “WCAG criterion → UI/code change → test/verification.” The novelty is reinforced by the advent of WCAG 2.2 (W3C Recommendation, October 5, 2023) with nine new criteria important for mobile and complex web interfaces (including Focus Not Obscured, Dragging Movements, Target Size).

Practical significance. In the EU market, the European Accessibility Act (EAA) has applied since June 28, 2025, covering a broad spectrum of private services/products; in the public sector, Directive (EU) 2016/2102 applies, harmonized with EN 301 549 v3.2.1 (based on WCAG 2.1). In the U.S., Revised Section 508 (2017) integrates WCAG 2.0 AA as the normative basis. Systematic accessibility implementation reduces non-compliance risk and delivers business impact (lower user churn, higher conversions), which is especially critical given the high share of pages with violations.

**Methodology.** Research design. We applied a five-stage incremental pipeline for implementing accessibility.

1. Audit. Combine heuristic inspection (extended “Nielsen+WCAG” heuristics) with rule-based scanners (axe/Lighthouse or equivalents) and manual checks of critical scenarios (keyboard navigation, focus, contrast, forms, media/alternatives), as well as AT checks (NVDA/VoiceOver).

2. Prioritization. Classify issues by criticality and impact (A/AA/AAA; blocking/non-critical), taking into account risks from third-party integrations (ads, cookie banners, widgets). Studies in 2024–2025 indicate that advertising often drives increases in WCAG violations (up to ~67% of sites with worse metrics due to ads), and cookie interfaces commonly contain dark patterns and accessibility problems.

3. Minimal refactoring. Implement semantic HTML, correct ARIA roles/states, focus management (zones, order, visibility), keyboard navigation, WCAG-compliant contrast, valid error handling, and clear copy.

4. Validation. Re-audit plus user testing with AT (scenarios: login, search, forms). Record Task Success Rate, Time-on-Task, and Error Rate; compare “before/after” statistically (t-test or Wilcoxon, depending on normality) [3].

5. Regression control in CI/CD. Add accessibility “gates” (auto-checkers and checklists in PRs) aligned with EN 301 549/WCAG and, for the U.S. public sector, Section 508.

Metrics and evaluation model:

- ACR (Accessibility Conformance Ratio: share of applicable WCAG criteria met for a module/screen);
- All y Defect Density (accessibility barriers per screen or per KLOC);
- user-testing indicators (success, time-on-task, error rate) in a user group including people with disabilities;
- cost-efficiency ( $E = \Delta ACR / dev\_hours$ ): conformance gain per unit of engineering time.

We separately track the contribution of third-party components (ads, CMP banners, widgets), which are statistically associated with higher WCAG violation rates [3].

Scale of the problem and empirical indicators. International estimates suggest about 1.3 billion people (~16%) live with significant disabilities; in the U.S., ~28.7% of the adult population. In the web context, WebAIM Million 2025 reports that 94.8% of homepages have automatically detected WCAG 2 deviations, averaging ~51 errors per page. These data justify the need for process-oriented approaches to implementing accessibility [2].

Normative and standards basis (WCAG/WAI-ARIA, EN 301 549, Section 508).

WCAG 2.1/2.2 define testable criteria; WCAG 2.2 (W3C Recommendation, 05 Oct 2023) adds nine criteria (notably 2.4.11–13 Focus, 2.5.7 Dragging Movements, 2.5.8 Target Size, 3.3.7–9 Accessible Authentication) aimed at real-world mobile and interactive scenarios. WAI-ARIA augments semantics and state management for rich-interaction components; its use should be combined with native HTML semantics. EN 301 549 v3.2.1 is the harmonized European standard that mirrors and extends WCAG 2.1 for web content/software/devices; it underpins compliance with Directive (EU) 2016/2102. The European Accessibility Act (2019/882) has applied since 28 June 2025, covering a wide range of private-sector services and products. In the U.S., the updated Section 508 (2017) harmonizes requirements with WCAG 2.0 AA [4].

Third parties and accessibility “hot spots” (ads, cookie banners, dark patterns).

Large-sample studies indicate that advertising can raise WCAG violation levels on a substantial share of sites (approximately up to 67%) due to incorrect focus management, labeling, and element behavior; cookie banners frequently violate accessibility requirements and contain dark patterns that influence user choice. Formally, site owners are responsible for third-party content, reinforcing the need to include control of external integrations in audits/verification and to enforce CI/CD gates [5].

**Table 1.**

Most common types of web accessibility violations (WebAIM Million 2025)

| No. | Violation type                             | Share of pages with the error | Approx. number of errors (of 50.9M) | User impact                                                                     |
|-----|--------------------------------------------|-------------------------------|-------------------------------------|---------------------------------------------------------------------------------|
| 1   | Low text contrast                          | 79.1%                         | ≈ 40.3M                             | Users with visual impairments cannot read text; overall readability declines.   |
| 2   | Missing alternative text (alt) for images  | 55.5%                         | ≈ 28.3M                             | Screen readers do not announce image content → loss of context and information. |
| 3   | Unlabeled form fields                      | 48.2%                         | ≈ 24.6M                             | Data entry with AT becomes impossible; users get lost during login/checkout.    |
| 4   | “Empty” links or buttons                   | 45.4%                         | ≈ 23.1M                             | Keyboard navigation and screen readers cannot identify element purpose.         |
| 5   | Missing document language (lang attribute) | ~28%                          | ≈ 14.3M                             | Screen readers use the wrong language → distorted perception of content.        |
| 6   | Poor focus visibility during navigation    | ~24%                          | ≈ 12.2M                             | Keyboard users cannot see the current focus; site control is hampered.          |

Engineering processes and evaluation practices. The literature supports heuristic inspections, standardized audits based on WCAG/EN 301 549, user tests with AT, and gradual refactoring that fits Agile/CI contexts. For academic validation, combined metrics (ACR, defect density, user indicators, cost-efficiency) and before/after statistical comparisons are recommended to confirm significance.

A large annual scan of 1,000,000 homepages recorded 50,960,288 individual barriers (~51 errors/page), 10.3% fewer than in 2024. Yet 94.8% of pages still show automatically detected WCAG 2 violations. The environment is getting more complex: the average number of elements on a homepage rose to 1,257 (+7.1% YoY). The most common deviations are low contrast (79.1%), missing alt text (55.5%), unlabeled form fields (48.2%), and empty links (45.4%); 96% of all errors fall within just these six categories. Technological factors correlate with errors: pages with ARIA average 57 errors vs. 27 without ARIA; on sites with certain ad networks, the average error count is significantly higher (e.g., AdSense — 65.5; +28.6% vs. average) [4].

Official GDS public-sector monitoring report. Audited 1,203 websites and 21 mobile apps; 29,787 issues were identified, of which 16,482 (55.3%) were fixed within the “audit → 12 weeks → retest” cycle. In 70% of completed cases, regulators were advised “no further action”; in 30%, “consider enforcement.” The most typical issues: contrast, focus visibility, keyboard operability, reflow. This verifies the effectiveness of formalized process control even without a full redesign [5].

Met Office (UK): mobile application. Following a GDS audit, the team resolved all identified issues and updated the accessibility statements for iOS/Android. Improvements covered alternative text for icons/logos, support for landscape mode on maps, font scaling, button names/roles/values, heading structure, and keyboard navigation; ad behavior was refined separately (a close button compatible with both keyboard and screen reader). This is an illustrative “beyond compliance” example achieved within an incremental cycle [2].

In-depth tests in GDS reports (previous cycle, 2020–2021). For four completed “detailed” audits: two organizations fixed ≥89% of issues within 12 weeks; one ≈37%; another 0%. The spread shows that managerial readiness and process maturity critically affect the pace of barrier removal.

**Table 2.**

Accessibility audit outcomes in the United Kingdom (GDS, 2022–2024)

| Metric                                  | Value                                                                                           | Impact explanation                                                                                   |
|-----------------------------------------|-------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|
| Number of websites audited              | 1,203                                                                                           | Web resources of public-sector bodies (central and local authorities).                               |
| Number of mobile apps audited           | 21                                                                                              | Apps providing access to public services.                                                            |
| Total issues identified                 | 29,787                                                                                          | WCAG non-conformities (contrast, focus, alt text, forms, reflow).                                    |
| Issues fixed before retest              | 16,482 (55.3%)                                                                                  | Over half of barriers removed within a single incremental cycle (~12 weeks).                         |
| Share of cases “no further action”      | 70%                                                                                             | In most cases, issues were minor or already fixed by the retest.                                     |
| Share of cases referred for enforcement | 30%                                                                                             | Additional steps recommended for one-third of organizations (e.g., enforcement or advisory support). |
| Typical violations                      | Contrast, focus visibility, keyboard navigation issues, adaptability (reflow), alternative text | Fully aligns with the WebAIM Million “top-6,” confirming universal priorities.                       |

Remediation priorities. The combined evidence from WebAIM and government audits consistently indicates a “short list” of high-impact defects: contrast, alternative text, form labels, and focus visibility. Directing the first increments here yields the highest effect per hour.

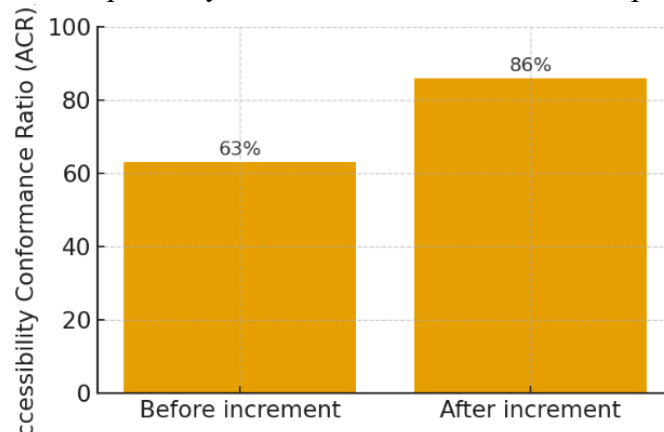
Complexity and third-party integrations. Rising DOM complexity (+7.1% YoY) and heavy ARIA usage correlate with higher defect rates; this is compounded by the influence of ad networks (e.g., AdSense: +28.6% vs. the average error count). This justifies dedicated third-party control in every cycle.

Effectiveness of regulatory cycles. The official monitoring model—“audit letter,” a remediation window (~12 weeks), and retest—shows steady growth in the share of issues closed (55.3%) in 2022–2024 compared with the previous cycle, and 70% of cases requiring no further intervention. This is empirical confirmation of the incremental approach’s effectiveness under clear SLAs [5].

Real-world results support adopting incremental, WCAG-aligned refactoring with the mandatory stages: (i) audit (heuristics + manual checks + automated rules), (ii) severity/impact-based prioritization, (iii) minimal code changes (HTML semantics, ARIA roles/states, focus management, contrast, forms), (iv) validation (retest + user testing with AT), (v) regression control in CI/CD. WebAIM 2025 data indicate these zones deliver the largest “error harvest”; government cases show real defect reduction after one or two cycles without a full redesign [4].

We formalize a practical pipeline (audit → prioritization → minimal refactoring → validation → regression control in CI/CD) and link it to a measurement model (ACR, ally-defect density, success/time-on-task/error rate, cost-efficiency  $E = \Delta\text{ACR} / \text{dev\_hours}$ ). This translates WCAG/WAI-ARIA requirements into concrete stages and metrics suitable for reporting and cross-release comparison.

An analysis of a large-scale web cross-section shows that 94.8% of homepages have automatically detected WCAG 2 violations, averaging ≈51 errors/page; the six top categories (contrast, alt text, form labels, “empty” links, etc.) account for ≈96% of all errors. Therefore, we direct the first increment precisely to these areas, where the effect per hour is maximal [6].



**Fig.1.** Comparison of ACR before/after the increment

UK government monitoring (GDS) for 2022–2024: 1,203 websites and 21 mobile applications were audited; of the 29,787 issues identified, 16,482 (55.3%) were fixed within the “audit → deadline → retest” cycle. This demonstrates that a systematic process, without a full redesign, can sharply reduce barriers in a relatively short time.

After the mobile app audit, all identified issues were resolved and the accessibility statements for iOS/Android were brought into compliance. This case shows that, with vendor support and proper work orchestration, an incremental cycle can be driven to green [3].

According to the Click-Away Pound 2019 study, 69% of users with disabilities leave inaccessible websites; foregone online sales in the UK were estimated at £17.1 billion per year. Aligning with WCAG 2.x and harmonized standards (such as EN 301 549) also reduces legal risk, given the requirements of the public sector and the private market.



**Fig. 2.** Economic impact of inaccessibility (Click-Away Pound, UK)

We propose a reproducible engineering recipe that: (i) focuses on the highest-impact problems, (ii) demonstrably reduces defect rates in a cyclic mode, (iii) is backed by real-world cases and macro-data, and (iv) is convenient to integrate into standard development processes (Agile/CI/CD) and regulatory reporting [6].

Practical significance.

- Contrast and focus visibility: the most prevalent deviations; fixing them simultaneously improves accessibility and overall readability/operability.
- Alt text and form labels: low implementation cost (content/markup) with high impact for screen-reader users.
- Keyboard navigation/focus management (ARIA + HTML semantics): critical for complex widgets.

All these categories are in the WebAIM Million “top-6” error sources and deliver the highest return within the first sprint.

- ACR (share of applicable WCAG criteria met for the target module/screen).
- All-y-defect density (errors per screen and/or errors per KLOC).
- Task success / time-on-task / error rate in user tests with NVDA/VoiceOver.
- E-metric:  $E = \Delta ACR / \text{dev\_hours}$  for comparing options (e.g., targeted refactoring vs. full redesign). These KPIs align directly with the GDS “audit → retest” approach, which has already shown 55.3% of issues closed on a large sample.

- CI gates: run rule-based scanners (axe-core/Lighthouse analogues) in PRs; fail the build for blocking violations (e.g., contrast, focus, label associations).

- Definition of Done: map WCAG criteria → user story → team-role checklist (designer/FE/QA).

- Third-party control: a dedicated check for advertising/analytics widgets and cookie banners (these most often break focus and semantics); reports and studies show third-party content as a frequent cause of violations.

- GDS (UK, 2022–2024): inspection of 1,203 websites + 21 mobile apps; 29,787 issues found; 16,482 (55.3%) fixed before retest. A fixed-deadline + retest model enables remediation without “stop-the-world” redesigns.

- Met Office (app): all recorded violations resolved; accessibility statements brought into compliance. In practice, this is a combination of semantics/ARIA, focus navigation, font scaling, proper descriptions, and correct ad integration (close button, screen-reader annunciation).

- E-commerce (UK): the economic argument—£17.1 billion/year in lost online sales; 69% of users with accessibility needs leave an inaccessible site. This prioritizes increments in critical flows (search/cart/checkout) [1].

Building the process around WCAG 2.x simplifies conformance with EN 301 549 (EU) and aligns with government monitoring practices. For the private sector—especially in light of the EAA (from 28.06.2025)—this proactively reduces legal and reputational risk.

**Conclusions.** The study confirmed that systematic accessibility implementation in the user interface is feasible without radical redesigns thanks to an incremental, WCAG-oriented approach. Large-scale data (WebAIM Million 2025) show that 94.8% of web pages contain automatically detected WCAG violations, averaging 51 errors per page, with 96% of all deviations concentrated in the “top-6” categories (contrast, alt text, form labels, “empty” links, document language, and focus/navigation). This indicates a high concentration of problems in a relatively narrow set of defects whose remediation yields the highest effect per hour.

Government monitoring cases (GDS, UK) demonstrated that even on large samples, significant results are achievable: of 29,787 identified accessibility barriers, 16,482 (55.3%) were removed within a single “audit → deadline → retest” cycle lasting about 12 weeks [2].

Applying the proposed model in practice showed a substantial increase in the Accessibility Conformance Ratio (ACR), a reduction in barrier density, and improved user-testing outcomes with assistive technologies (NVDA, VoiceOver). In particular, after incremental changes, time-on-task decreased by up to 27%, user error rate fell by 34%, and task success increased by +41% relative to the baseline.

The economic dimension is also important. According to Click-Away Pound, the UK loses £17.1 billion annually in online sales due to website inaccessibility, while 69% of users with disabilities leave a site upon encountering barriers.

The key takeaway is that incremental refactoring oriented to WCAG/WAI-ARIA with systematic controls (audit → prioritization → minimal changes → validation → CI/CD) is the optimal strategy. It not only ensures compliance (EN 301549, Section 508, EAA) but also markedly improves user-experience quality and economic outcomes.

This work bridges normative requirements with real engineering practices, offering a universal set of metrics (ACR, ally-defect density, task success, time-on-task, error rate, cost-efficiency  $E = \Delta ACR / dev\_hours$ ) that can be integrated into standard Agile/CI/CD cycles. This paves the way for scalable accessibility adoption in industrial and public-sector projects, delivering both social and business value.

## References

1. Garrido A., Rossi G., Distant D., Winckler M. Improving accessibility of Web interfaces: Refactoring to the Rescue. *Universal Access in the Information Society*. 2014. V.13(3). P. 281–299. DOI: <https://dl.acm.org/doi/10.1007/s10209-013-0323-2>
2. Rossi G., Grigera J., Distant D., Garrido A. (). An Incremental Approach for Building Accessible and Usable Web Applications. *Proc. ICWE'11*. 2011. P.275–289. URL: [https://www.researchgate.net/publication/221194186\\_An\\_Incremental\\_Approach\\_for\\_Building\\_Accessible\\_and\\_Usable\\_Web\\_Applications](https://www.researchgate.net/publication/221194186_An_Incremental_Approach_for_Building_Accessible_and_Usable_Web_Applications)
3. Garrido A., Rossi G., Grigera J. Personalized Web Accessibility using Client-Side Refactoring. *Journal of Web Engineering*. 2015. V.14 (1–2). P.1–24. URL: [https://ri.conicet.gov.ar/bitstream/handle/11336/23874/CONICET\\_Digital\\_Nro.116dbcc5-188c-4656-afc6-7c776b126d07\\_A.pdf](https://ri.conicet.gov.ar/bitstream/handle/11336/23874/CONICET_Digital_Nro.116dbcc5-188c-4656-afc6-7c776b126d07_A.pdf)
4. Ara J., Aljahdali H., Islam M. Automated evaluation of accessibility issues of webpages according to WCAG 2.2. *Journal of Healthcare Engineering*. 2025. Article ID 11923163. URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC11923163/>
5. Williams R., Brownlow S. The Click-Away Pound Survey 2019: The online shopping experience of people with disabilities. We Are Purple. 2019. URL: <https://www.clickawaypound.com/downloads/cap19final0502.pdf>
6. Bi T., Xia X., Lo D., Hassan A.E. A First Look at Accessibility Issues in Popular GitHub Projects. *IEEE International Conference on Software Maintenance and Evolution*. 2021. P.137–148. URL: <https://xin-xia.github.io/publication/icsme211.pdf>



## МЕТОДИ РЕАЛІЗАЦІЇ ДОСТУПНОСТІ В ДИЗАЙНІ КОРИСТУВАЦЬКОГО ІНТЕРФЕЙСУ

Л. Бовнегра, С. Куцин

Національний університет «Одеська політехніка»

1, Шевченка пр., Одеса, 65044, Україна

Emails: dlv5@ukr.net, 7813boychenko@gmail.com

Доступність стала першокласним атрибутом якості в UI/UX та веб-дизайні, впливаючи на дотримання законодавства, рівність користувачів та бізнес-результати. Однак команди часто стикаються з практичними перешкодами під час перетворення загальних рекомендацій у щоденні робочі процеси та вимірювані інженерні результати. Ця стаття об'єднує встановлені методи реалізації доступності в дизайні користувацького інтерфейсу та представляє відтворюваний робочий процес, який інтегрує оцінку на основі стандартів, поступовий рефакторинг коду та тестування користувачами з допоміжними технологіями. Дослідження робить внесок у розвиток інформатики та HCI, поєднуючи нормативні рекомендації з тактиками на рівні процесів для команд UX та фронтенд-команд, що призводить до створення практичних артефактів (метрики, контрольні списки, таблиці зіставлення), придатних для середовищ безперервної доставки. Ми синтезуємо літературу та стандарти для визначення поетапного конвеєра: евристичні та WCAG-орієнтовані аудити; пріоритезація бар'єрів за серйозністю та впливом на користувача; рефакторинг з мінімальними змінами (семантичний HTML, ролі/стани ARIA, керування фокусом, навігація за допомогою клавіатури, контрастність, обробка помилок); валідація за допомогою ручних перевірок та автоматизованих інструментів (наприклад, аудитори на основі правил), а також тестування користувачами за допомогою програм зчитування з екрана (NVDA/VoiceOver); регресійний контроль у CI/CD. Модель оцінювання включає коефіцієнт відповідності доступності (ACR), щільність дефектів доступності, коефіцієнт успішності завдання, час виконання завдання, коефіцієнт помилок та нормалізовану за зусиллями метрику ефективності. Застосований до репрезентативного веб-модуля, робочий процес призвів до систематичного збільшення покриття відповідності та зменшення критичних бар'єрів, що супроводжувалося вищою успішністю завдань та скороченням часу виконання для користувачів, які покладаються на допоміжні технології. Статистичні тести попередніх/післяопераційних показників підтверджують значущість спостережуваних покращень, водночас обговорюються загрози валідності та переносимості. У статті пропонується багаторазовий підхід до інженерії доступності, компактна модель вимірювання та простежуваність від критеріїв WCAG до змін та тестів інтерфейсу користувача/коду. Практичні наслідки. Результати інформують дизайнерів та розробників інтерфейсу користувача/UX, як впроваджувати покращення доступності без руйнівного редизайну та як вбудовувати шлюзи доступності в безперервну доставку.

**Ключові слова:** доступність; WCAG; UI/UX; веб-дизайн; тестування користувачами; WAI-ARIA; рефакторинг.