бути окремим завданням і стає природною частиною обговорень [2]. Ще одним практичним моментом є створення маленьких навчальних мінісесій усередині команди. Наприклад, один учасник може показати просту типову помилку і коротко пояснити, як її уникати. Це не займає багато часу, але сприяє тому, що команда росте у цьому напрямі разом [5].

Результати аналізу показали, що такі практичні кроки працюють найбільш ефективно тоді, коли повторюються регулярно. Маленькі, але постійні дії створюють відчутну різницю у якості коду і в стійкості продукту до можливих загроз [1].

Поєднання гнучких методів і безпеки можливе, якщо додавати до процесу маленькі, але регулярні практичні дії. Авторські ідеї у вигляді короткої підготовчої перевірки, створення журналу помилок, обговорення простих порад під час зустрічей і маленьких внутрішніх навчань допомагають сформувати культуру уважності. Такий підхід робить продукт надійнішим і не сповільнює роботу команди, що дає змогу зберегти переваги гнучкої методології.

Список використаних джерел

1. Williams L. Smith Br. Integrating Security Practices into Agile Development Processes. *IEEE Software.* 2023. URL: https://doi.org/10.1109/MS.2023.0145

2. Baca D., Boldt M. Security Training in Agile Software Development Teams. *Information and Software Technology.* 2024. URL: https://www.sciencedirect.com/science/article/pii/S0950584924001123 (дата звернення: 01.12.2025).

3. McGraw G. Software Security: Building Security. *Addison-Wesley Professional.* 2020. URL: https://www.informit.com/store/software-security-building-security-in-9780321356703 (дата звернення: 01.12.2025).

4. Alshammari B., Singh H. Secure Agile Development Framework for Modern Software Projects. *International Journal of Secure Software Engineering.* 2022. DOI: 10.4018/IJSSE.20220701.

5. Ghanbari H., Shahriari H. Embedding Security Activities into Agile Methods: A Systematic Review. *Journal of Systems and Software.* 2025. URL: https://www.journals.elsevier.com/journal-of-systems-and-software (дата звернення: 01.12.2025).

УДК 004.5

Афанасьєва Д. С., здобувачка 4 курсу спеціальності 122 Комп'ютерні науки,
Веселовська Н. Р., д-р техн. наук, професор, професор кафедри інформаційних технологій

АЛГОРИТМІЧНІ МЕТОДИ АДАПТАЦІЇ ВЕБКОНТЕНТУ ДЛЯ ЛЮДЕЙ З ВАДАМИ ЗОРУ

Донецький національний університет імені Василя Стуса, м. Вінниця

Зі зростанням складності вебінтерфейсів і перенесенням ключових життєвих процесів у цифровий простір питання доступності сайтів для людей з порушен-

нями зору перестає бути вузькоспеціальним. Стандарти на кшталт WCAG 2.2 задають чіткі вимоги до контенту [1], проте значна частина вебресурсів і надалі не відповідає навіть базовим критеріям: аналіз одного мільйона головних сторінок WebAIM у 2025 р. показав, що понад 94 % з них мають автоматично виявлені порушення WCAG із середнім значенням понад 50 помилок на сторінку [2]. Це означає, що покладатися лише на правильну розробку не можна – потрібні алгоритмічні рішення, які здатні аналізувати вже наявний вебконтент і адаптувати його під потреби користувача з вадами зору. Мета роботи – показати, як принципи доступності можуть бути реалізовані через алгоритмічні методи, що працюють у режимі реального часу: від евристичного аналізу DOM і CSS до використання моделей машинного навчання для персональної адаптації.

Для користувача з повноцінним зором поганий контраст чи дрібний шрифт – лише незручність. Для людини зі слабким зором це може бути фактичною відсутністю доступу до послуги: неможливістю прочитати інструкцію, заповнити форму, підтвердити дію. У таких умовах логічним кроком є перехід від ручного забезпечення доступності до алгоритмічних рішень. Ідеться не тільки про автоматизоване тестування, яке виявляє порушення стандартів, а й про адаптацію інтерфейсу в режимі реального часу: коли програмний модуль аналізує вже розміщену сторінку й автоматично підлаштовує її під потреби конкретного користувача з вадами зору.

Якщо розглядати адаптивну систему з практичного боку, її реалізація складається з кількох логічних шарів. На першому етапі здійснюється збір даних: модуль отримує доступ до DOM-дерева сторінки, аналізує атрибути елементів та їх стилі з урахуванням правил наслідування. Система ідентифікує текстові вузли, заголовки, посилання, кнопки, поля форм, зображення та інші значущі для користувача сутності. Далі формується аналітичний рівень, на якому обчислюється контрастність тексту і фону, аналізуються розміри шрифту, міжрядкові інтервали й відстані між блоками, перевіряється наявність альтернативного тексту в зображення, структура заголовків, доступність елементів з клавіатури та видимість фокуса, а також враховуються попередні користувацькі налаштування. На трансформаційному рівні система реалізує результати аналізу, динамічно змінюючи CSS-властивості, додаючи або коригуючи ARIA-атрибути, перерозподіляючи блоки та змінюючи порядок фокусування. Технічно це часто виконується через підключення додаткового стильового шару з вищим пріоритетом без втручання у вихідний код сайта. Завершальний рівень пов’язаний із взаємодією з користувачем і передбачає панель налаштувань, де задаються параметри шрифту, кольорової схеми, контрастності й анімації; ці значення зберігаються локально або в профілі користувача та надалі використовуються для персоналізованої адаптації.

Найпростішим практичним варіантом реалізації такої системи є правилоорієнтований підхід, у межах якого адаптація інтерфейсу здійснюється на основі заздалегідь визначених перевірок і трансформацій. Це виглядає як набір функцій, кожна з яких відповідає за конкретний аспект доступності: нормалізацію шрифтів, перевірку контрастності кольорів, підвищення видимості фокуса, корекцію структури заголовків або доступності елементів керування. Такі алгоритми не потребують складних обчислень, легко піддаються тестуванню та верифікації,

а їх результат є передбачуваним. У практичному застосуванні цього вже достатньо, щоб суттєво покращити зручність користування сайтами для людей із низьким зором, особливо на тих ресурсах, де вимоги WCAG взагалі не були враховані на етапі розробки.

Водночас правилоорієнтований підхід має природні обмеження, адже він працює переважно з формальними параметрами інтерфейсу і майже не враховує реальну поведінку користувача та контекст використання. Саме тому останнім часом поширення набувають гібридні методи, у яких базові правила поєднуються з аналізом даних [3]. Зібрані в процесі взаємодії агреговані поведінкові характеристики, як-от швидкість навігації, частота помилкових кліків, затримка фокуса на окремих елементах, частота зміни масштабу або режиму контрасту, дають змогу уточнювати початкові налаштування інтерфейсу. На основі цих даних система може поступово адаптувати базові алгоритми під індивідуальні потреби користувача, переходячи від універсальної адаптації до персоналізованої.

У реальному використанні це виглядає як поетапний процес. Під час першого відвідування сайт адаптується відповідно до стандартного набору правил, забезпечуючи мінімально необхідний рівень зручності: збільшений шрифт, підвищений контраст, чіткий фокус елементів керування. Паралельно система фіксує узагальнені дані про характер взаємодії користувача з інтерфейсом, не втручаючись у його роботу. Надалі, використовуючи ці спостереження, алгоритмічний модуль може коригувати первинні налаштування, наприклад, збільшувати базовий розмір тексту, зменшувати кількість одночасно відображених елементів або змінювати логіку відображення меню, роблячи її простішою для конкретного користувача.

Окремим напрямом розвитку є використання глибинного навчання для аналізу візуальної структури вебсторінок і складніших патернів взаємодії. У цьому випадку система працює не лише з числовими характеристиками стилів, а й із цілісним образом сторінки, отриманим, наприклад, зі скриншота. Нейронна мережа може виявляти проблемні зони з погляду сприйняття, як-от перевантажені інформацією блоки, дрібний текст або надмірно контрастні декоративні елементи, після чого формуються рекомендації щодо переформатування інтерфейсу [4]. Хоча повноцінна інтеграція таких моделей безпосередньо у браузер є технічно складною, на практиці можливе використання серверної обробки з передачею клієнту вже готових результатів адаптації, що є цілком прийнятним рішенням для навчальних і прикладних проєктів.

Отже, алгоритмічні методи адаптації вебконтенту для людей з порушеннями зору забезпечують перехід від формального декларування доступності до практичного аналізу та динамічної зміни інтерфейсу під конкретного користувача. На технічному рівні це реалізується через багатошарові програмні модулі, які аналізують структуру та стилі сторінки, застосовують правила WCAG 2.2 і за потреби доповнюють їх моделями машинного навчання для врахування поведінкових даних. Поєднання правилоорієнтованих і ML-підходів дає змогу створювати гібридні системи персоналізованої доступності, що можуть функціонувати як браузерні розширення, вбудовані модулі або інструменти для аудиту й покращення інтерфейсів, а з урахуванням посилення регуляторних вимог і стабільної

поширеності порушень доступності цей напрям набуває чітко вираженої практичної значущості.

Список використаних джерел

1. Web Content Accessibility Guidelines (WCAG) 2.2. *W3C Recommendation.* URL: https://www.w3.org/TR/WCAG22/ (дата звернення: 29.11.2025).

2. The 2025 Report on the Accessibility of the Top 1,000,000 Home Pages. *The WebAIM Million.* URL: https://webaim.org/projects/million/ (дата звернення: 30.11.2025).

3. Machine Learning Driven Adaptive Accessibility Interfaces for Enhanced User Experience / S. Patel, R. Mendoza, I. Cruz, N. Kim, P. Singh, A. Martinez. URL: https://doi.org/10.13140/RG.2.2.12545.36965 (дата звернення: 27.04.2026).

4. Deep Learning-Enhanced Adaptive Interface for Improved Accessibility in E-Government Platforms / X. Li; Y. Wang et al. *Proceedings of the 6th International Conference on Frontier Technologies of Information and Computer (ICFTIC 2024).* China, 2024. P. 1692–1695. URL: https://doi.org/10.1109/ICFTIC64248.2024.10912984 (дата звернення: 27.04.2026).

УДК 004.428.2

Дикий Д. С., здобувач 2 курсу
ОС «Магістр» спеціальності
F3 Комп'ютерні науки,
Потапова Н. А., канд. екон. наук, доцент,
доцент кафедри інформаційних технологій

ВЕБПОРТАЛ УПРАВЛІННЯ ЛОГІСТИКОЮ МАГАЗИНУ КОМПЛЕКТУЮЧИХ СМАРТСИСТЕМ

Донецький національний університет імені Василя Стуса, м. Вінниця

Сучасні інформаційні системи дедалі частіше будуються на основі принципів гнучкості, масштабованості та інтегрованості, оскільки традиційні програмні рішення вже не здатні ефективно задовольняти потреби бізнесу в умовах постійного зростання обсягів даних та високої варіативності користувацьких сценаріїв. Особливо це актуально для систем, що забезпечують підтримку логістичних процесів, де швидкість обробки інформації, стабільність роботи та можливість адаптації до змін є критично важливими.

Створення вебпорталу для автоматизації логістики магазину комплектуючих для смартсистем потребувало формування архітектури, яка б поєднувала високу продуктивність, простоту супроводу та можливість гнучкого масштабування. Важливо було забезпечити чітке структурування компонентів, стабільність обміну даними, а також здатність до інтеграції зі сторонніми сервісами та інструментами.

Архітектура розробленого вебпорталу ґрунтується на принципах клієнт-серверної взаємодії та патерні Model-View-Controller (MVC), що дає змогу чітко