

Воронцов Глеб Игоревич, бакалавр
Уфимский университет науки и технологий
Vorontsov Gleb Igorevich
Ufa University of Science and Technology

МЕТОДЫ ОБЕСПЕЧЕНИЯ ДОСТУПНОСТИ ВЕБ-ИНТЕРФЕЙСОВ НА ОСНОВЕ ТРЕБОВАНИЙ WCAG 2.2 METHODS FOR ENSURING WEB INTERFACE ACCESSIBILITY BASED ON WCAG 2.2

Аннотация. Рассмотрены методы проектирования и проверки доступных веб-интерфейсов на основе WCAG 2.2. Описаны семантическая разметка, клавиатурное управление, применение WAI-ARIA и контроль динамических состояний. Предложен комбинированный процесс тестирования, включающий автоматические, ручные и пользовательские проверки.

Abstract. Methods for designing and evaluating accessible web interfaces based on WCAG 2.2 are reviewed. Semantic markup, keyboard operation, WAI-ARIA usage, and dynamic state control are described. A combined testing process involving automated, manual, and user-based evaluation is proposed.

Ключевые слова: Веб-доступность, WCAG 2.2, WAI-ARIA, семантическая разметка, клавиатурная навигация, тестирование.

Keywords: Web accessibility, WCAG 2.2, WAI-ARIA, semantic markup, keyboard navigation, testing.

Введение

Доступность веб-интерфейса означает возможность воспринимать содержимое и выполнять действия при различных сенсорных, моторных и когнитивных особенностях пользователя. Она необходима людям, использующим экранные дикторы, клавиатуру без мыши, увеличение, голосовое управление или альтернативные устройства ввода. Недоступный компонент может быть визуально понятным, но не иметь программного имени, терять фокус либо сообщать изменение только цветом. Поэтому доступность определяется совместной работой дизайна, разметки, сценариев и тестирования.

Масштаб проблемы подтверждает исследование WebAIM Million за 2025 год: автоматически обнаруживаемые нарушения WCAG присутствовали на 94,8 % исследованных главных страниц, а среднее число ошибок составляло 51 на страницу [4]. Наиболее частыми были недостаточный контраст, отсутствие альтернативного текста и подписей полей. При этом автоматический анализ выявляет лишь часть барьеров. Цель статьи – систематизировать технические и процессные методы обеспечения доступности динамических веб-интерфейсов на основе WCAG 2.2.

Нормативная модель WCAG 2.2

WCAG 2.2 группирует требования вокруг четырех принципов: содержимое должно быть воспринимаемым, управляемым, понятным и надежным для интерпретации пользовательскими агентами [1]. Критерии успеха сформулированы независимо от конкретной технологии и разделены по уровням А, АА и ААА. Соответствие уровню АА предполагает выполнение всех критериев А и АА для полной страницы, включая ее состояния и адаптивные варианты. Поэтому проверка отдельного компонента полезна, но не доказывает соответствие всего пользовательского маршрута.

Восприимчивость начинается с текстовых альтернатив изображениям, подписей для мультимедиа, достаточного контраста и возможности увеличивать содержимое без потери функций. Информация не должна зависеть только от цвета, положения или формы. При адаптивной верстке важно сохранять логический порядок и исключать горизонтальную прокрутку там, где критерий допускает перерасположение. Альтернативный текст описывает назначение изображения в контексте, а декоративный элемент должен быть корректно исключен из дерева доступности [1].



Управляемость требует доступа ко всем действиям с клавиатуры, заметного индикатора фокуса и предсказуемого порядка переходов. WCAG 2.2 дополняет требования к видимости фокуса, размеру цели и альтернативам жестам перетаскивания [1]. Понятность обеспечивается последовательной навигацией, явными подписями, конкретными сообщениями об ошибках и помощью при вводе. Критерий доступной аутентификации ограничивает ситуации, в которых от пользователя требуется решить когнитивную задачу без альтернативы.

Принцип надежности связывает интерфейс с браузерами и вспомогательными технологиями. Интерактивный элемент должен передавать программное имя, роль, значение и состояние. В динамическом приложении недостаточно визуальное изменение кнопки или вкладки: соответствующее состояние должно появиться в модели доступности, а важное обновление – быть объявлено подходящим способом. Требования сохраняются после клиентской маршрутизации, открытия модального окна, появления ошибки и асинхронного обновления данных.

Семантика и интерактивные компоненты

Предпочтительной основой является нативная HTML-семантика. Элементы `button`, `a`, `input`, `label`, `heading` и `landmark` уже имеют ожидаемые роли и клавиатурное поведение. Замена кнопки элементом `div` требует ручную воспроизвести фокус, активацию с клавиатуры, имя и состояния, что увеличивает риск расхождений. Заголовки образуют структуру документа, `landmarks` ускоряют навигацию, а связанная с полем подпись остается доступной при визуальном изменении макета. Семантика должна отражать назначение, а не внешний вид компонента.

Управление фокусом особенно важно для одностраничных приложений. После перехода пользователь должен понимать, что содержимое изменилось; при открытии диалога фокус переносится внутрь, остается в его логическом контексте и возвращается к иницилирующему элементу после закрытия. Нельзя создавать положительные значения `tabindex` для искусственного порядка, расходящегося с DOM. Видимый индикатор фокуса не скрывается ради дизайна, а интерактивные элементы сохраняют ожидаемые клавиши активации.

WAI-ARIA определяет роли, состояния и свойства для компонентов, семантика которых отсутствует в HTML [2]. Руководство APG описывает шаблоны вкладок, меню, диалогов и других виджетов вместе с клавиатурным взаимодействием [3]. ARIA не изменяет поведение автоматически: объявление `role=button` не добавляет обработку клавиш. Ошибочная роль способна скрыть нативную семантику или передать неверное состояние. Поэтому сначала выбирается подходящий HTML-элемент, затем добавляются только необходимые ARIA-атрибуты и проверяется фактическое объявление экранным диктором.

Комбинированный контроль качества

Автоматические проверки эффективны для массового поиска формализуемых ошибок: отсутствующего `alt`, неподписанного поля, некоторых нарушений контраста, недопустимых сочетаний ролей и повторяющихся идентификаторов. Формат ACT Rules 1.1 задает прозрачное описание области применимости, ожиданий и примеров тестового правила, что способствует воспроизводимости результатов [7]. Однако инструмент не может надежно определить смысл альтернативного текста, логичность фокуса или понятность сообщения. Исследование средств оценки показывает риск исключительной зависимости от автоматических тестов [5].

Ручная проверка включает прохождение всех функций только клавиатурой, увеличение масштаба, анализ перестроения макета, режим повышенного контраста и контроль динамических состояний. Экранный диктор используется для проверки имен, ролей, структуры заголовков, статуса форм и порядка чтения. Тестируется не только главная страница, но и репрезентативные маршруты: регистрация, поиск, изменение данных, ошибки,



пустые состояния и истечение сессии. Методология WCAG-EM предлагает определить область оценки, исследовать сайт, выбрать выборку страниц, выполнить проверку и оформить результаты [8].

Пользовательское тестирование выявляет барьеры, которые трудно вывести из формального критерия. В исследовании проблем слепых пользователей показано, что соблюдение руководств отражает лишь часть реального опыта взаимодействия [6]. Участники с различными способами доступа помогают оценить понятность последовательности действий, стоимость навигации и возможность восстановиться после ошибки. Такое тестирование не заменяет нормативный аудит, поскольку небольшая выборка не покрывает все критерии, но дополняет его сведениями о фактической пригодности интерфейса.

Заключение

Обеспечение доступности веб-интерфейса начинается с нормативной модели WCAG 2.2 и реализуется через семантический HTML, клавиатурное управление, понятные сообщения и корректную передачу динамических состояний. WAI-ARIA применяется для дополнения, а не замены нативной семантики. Особое внимание требуется к одностраничным приложениям, где маршрутизация и асинхронные обновления могут происходить без естественной перезагрузки документа и без автоматического уведомления пользователя.

Достоверная оценка требует сочетания автоматических правил, ручного аудита, проверки вспомогательными технологиями и участия пользователей с инвалидностью. Автоматизация ускоряет обнаружение повторяемых дефектов, но не подтверждает смысловую корректность и удобство сценария. Интеграция требований в дизайн-систему, компоненты, критерии приемки и регрессионный контроль делает доступность постоянным свойством процесса разработки, а не отдельной операцией перед публикацией.

Список литературы:

1. W3C. Web Content Accessibility Guidelines (WCAG) 2.2. W3C Recommendation, 12.12.2024. URL: <https://www.w3.org/TR/WCAG22/> (дата обращения: 11.08.2026).
2. W3C. Accessible Rich Internet Applications (WAI-ARIA) 1.2. W3C Recommendation, 06.06.2023. URL: <https://www.w3.org/TR/wai-aria-1.2/> (дата обращения: 11.08.2026).
3. W3C. ARIA Authoring Practices Guide. URL: <https://www.w3.org/WAI/ARIA/apg/> (дата обращения: 11.08.2026).
4. WebAIM. The WebAIM Million: The 2025 Report on the Accessibility of the Top 1,000,000 Home Pages. 2025. URL: <https://webaim.org/projects/million/2025> (дата обращения: 11.08.2026).
5. Vigo M., Brown J., Conway V. Benchmarking Web Accessibility Evaluation Tools: Measuring the Harm of Sole Reliance on Automated Tests // Proceedings of W4A 2013. 2013. DOI: 10.1145/2461121.2461124.
6. Power C., Freire A. P., Petrie H., Swallow D. Guidelines Are Only Half of the Story: Accessibility Problems Encountered by Blind Users on the Web // Proceedings of CHI 2012. 2012. P. 433-442. DOI: 10.1145/2207676.2207736.
7. W3C. Accessibility Conformance Testing (ACT) Rules Format 1.1. W3C Recommendation, 05.02.2026. URL: <https://www.w3.org/TR/act-rules-format/> (дата обращения: 11.08.2026).
8. W3C. Website Accessibility Conformance Evaluation Methodology (WCAG-EM) Overview. Updated 05.02.2026. URL: <https://www.w3.org/WAI/test-evaluate/conformance/wcag-em/> (дата обращения: 11.08.2026)

