



Impact of Accessibility-First Design Systems on WCAG 2.1 Compliance and Inclusive User Experience in Enterprise Software Platforms: A Mixed-Methods Evaluation

Gazi Yeasin Arifin¹;

[1]. Lead UX Designer, Kite Games Studio Ltd, Dhaka, Bangladesh.
Email: arifin.yeasin@gmail.com

Doi: [10.63125/nxh4gp19](https://doi.org/10.63125/nxh4gp19)

Received: 15 April 2023; Revised: 11 May 2023; Accepted: 05 June 2023; Published: 02 July 2023

Abstract

This quantitative cross-sectional comparative study examined the relationship between accessibility-first design-system practices and Web Content Accessibility Guidelines (WCAG) 2.1 conformance across enterprise software interfaces. The analysis included 60 enterprise platforms, comprising 30 accessibility-first and 30 conventional platforms, with 180 applications, 1,440 interface screens, 360 workflows, 4,860 reusable components, and 32,400 interface elements. Accessibility performance was assessed through overall WCAG 2.1 compliance, Level A and Level AA conformance, violation frequency and severity, POUR dimensions, color contrast, keyboard operability, screen-reader compatibility, and assistive-technology task success. Correlation, regression, independent-samples t-tests, Mann–Whitney U tests, and ANOVA were applied at a significance level of $p < .05$, with effect sizes and 95% confidence intervals used to determine practical significance. Accessibility-first platforms achieved an overall WCAG compliance rate of 91.8%, compared with 74.6% for conventional platforms, while Level AA conformance reached 88.2% and 68.5%, respectively. Conventional platforms recorded 17.51 violations per 100 interface elements compared with 6.90 among accessibility-first platforms. Perceivable violations represented 40.0% of all identified failures, followed by Operable violations at 27.0%. Accessibility-first platforms also achieved higher contrast compliance (96.1% vs. 79.4%), keyboard task success (92.4% vs. 70.8%), screen-reader compatibility (90.6% vs. 69.7%), and assistive-technology task success (88.9% vs. 66.4%). Accessible component reuse was strongly negatively correlated with normalized WCAG violation frequency ($r = -.71, p < .001$), while accessibility-oriented design-token adoption was strongly positively associated with contrast compliance ($r = .76, p < .001$). The regression model predicting normalized WCAG violations explained 64.8% of the variance, whereas the contrast-compliance model explained 67.2%. Overall WCAG compliance also demonstrated a very large group effect (Cohen's $d = 2.45$). The findings demonstrated that accessibility-first design systems, particularly accessible component reuse and standardized design tokens, were associated with substantially stronger technical and functional accessibility outcomes. The study highlighted the value of embedding accessibility within reusable enterprise interface infrastructure rather than relying primarily on application-level remediation.

Keywords

WCAG, Accessibility, Design Systems, Enterprise, Compliance.

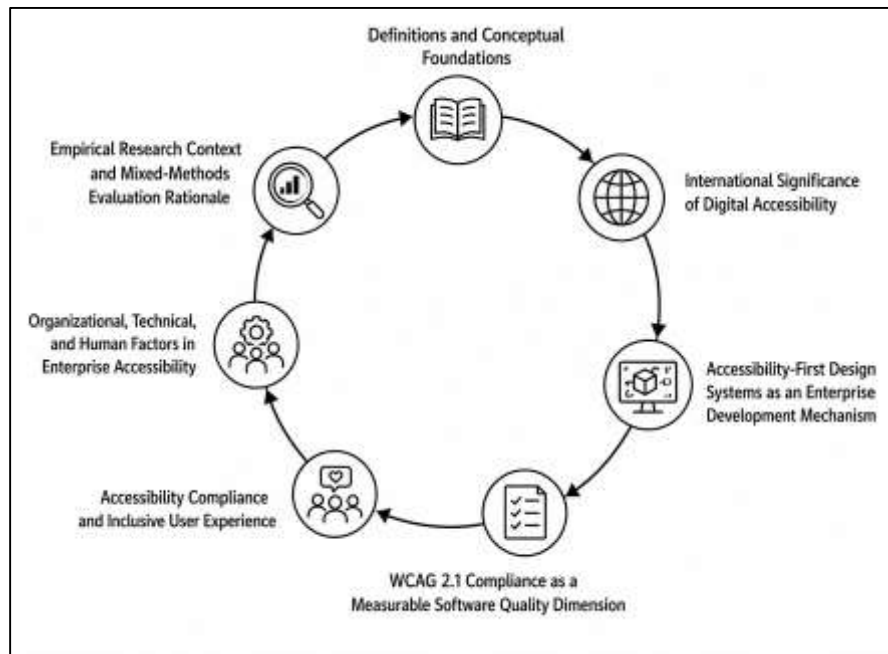
INTRODUCTION

Accessibility-first design can be defined as a systematic approach to digital product development in which accessibility requirements are treated as foundational design and engineering criteria rather than as corrective measures applied after an interface has been developed. Within enterprise software environments, an accessibility-first design system extends this principle across reusable interface components, design tokens, interaction patterns, documentation, coding conventions, and testing procedures so that accessibility becomes embedded in the architecture of the product ecosystem (Campoverde-Molina et al., 2021). A design system itself represents an organized collection of reusable components, standards, principles, and implementation resources that enables multiple teams to produce consistent digital interfaces at organizational scale. When accessibility is incorporated at this systemic level, components such as buttons, forms, navigation controls, dialogs, tables, notifications, menus, input fields, typography, color schemes, focus indicators, and error messages can be designed around predefined accessibility requirements before being reused across enterprise applications. Web accessibility, in turn, refers to the extent to which digital information, functionality, and interactions can be perceived, understood, navigated, and operated by people with diverse sensory, physical, cognitive, speech, and neurological characteristics. The Web Content Accessibility Guidelines 2.1 provide one of the principal international frameworks for operationalizing this concept by organizing accessibility around four fundamental principles: content and interfaces should be perceivable, operable, understandable, and robust (Noh et al., 2015). WCAG 2.1 further structures conformance through Levels A, AA, and AAA and addresses areas such as text alternatives, keyboard accessibility, color contrast, adaptable content, focus management, input assistance, navigation, semantic structure, and compatibility with assistive technologies. Inclusive user experience is broader than technical accessibility because it concerns the quality of interaction experienced across users with differing abilities, technologies, contexts, and interaction strategies. Research connecting accessibility with user experience has consequently examined not only technical violations but also usability, task performance, workload, aesthetics, perceived accessibility, satisfaction, and positive or negative experience. Enterprise software platforms create a particularly important setting for examining these relationships because their interfaces frequently contain complex workflows, data-intensive dashboards, forms, tables, authentication procedures, administrative controls, dynamic components, and interconnected applications (Sanchez-Gordon et al., 2019). Accessibility-first design systems therefore represent an organizational and technological mechanism through which individual accessibility requirements can be translated into reusable standards across extensive software environments, connecting interface architecture, WCAG conformance, software quality, and inclusive experience within a common analytical framework.

The international significance of accessibility-first digital design is grounded in the expanding role of software platforms in employment, education, commerce, public administration, financial services, healthcare administration, communication, and organizational decision-making. Access to digital systems increasingly determines whether individuals can participate independently and effectively in institutional processes, making inaccessible interfaces a significant form of digital exclusion (Chee & Weaver, 2021). WCAG has developed into an internationally recognized reference framework for addressing this problem and provides recommendations intended to improve access for people with blindness and low vision, deafness and hearing loss, mobility limitations, speech disabilities, photosensitivity, and combinations of disabilities, while also addressing some cognitive and learning requirements. The international relevance of these standards is reinforced by their use as technical reference points within accessibility policies, procurement requirements, institutional standards, and regulatory structures in different jurisdictions. Research across national and institutional contexts continues to document accessibility shortcomings. Evaluations of UK public-sector websites identified recurring non-conformance problems across a substantial sample of government-related sites, while investigations of Indian institutional websites have documented accessibility errors across WCAG conformance categories (Ali et al., 2017). Research examining corporate websites has similarly investigated the adoption of W3C accessibility requirements among hundreds of internationally prominent firms, linking accessibility with social inclusion and organizational responsibility. Studies of educational digital resources have also reported measurable WCAG 2.1 problems across major

online learning platforms, demonstrating that accessibility barriers occur within systems serving large and diverse populations. These findings position accessibility as a cross-sector concern rather than an isolated web-development issue. Enterprise software adds another international dimension because multinational organizations often deploy common applications, component libraries, cloud platforms, and digital workflows across employees and customers located in multiple legal and cultural environments (García-Santiago & Olvera-Lobo, 2021).

Figure 1: Accessibility-First Enterprise Design Systems



Organizations therefore require mechanisms capable of translating accessibility principles into repeatable interface practices rather than resolving accessibility defects separately within every application. Large institutional design systems illustrate this systemic approach. The U.S. Web Design System explicitly incorporates accessibility testing into reusable components, while government design systems in the United Kingdom and Scotland integrate WCAG-oriented requirements into component guidance and development practices. Enterprise technology providers have likewise positioned accessibility within design-system architecture, including accessible color, contrast, and interaction requirements. Accessibility-first design systems consequently occupy an internationally significant position between global accessibility standards and the operational realities of large-scale software production (Nacheva, 2021).

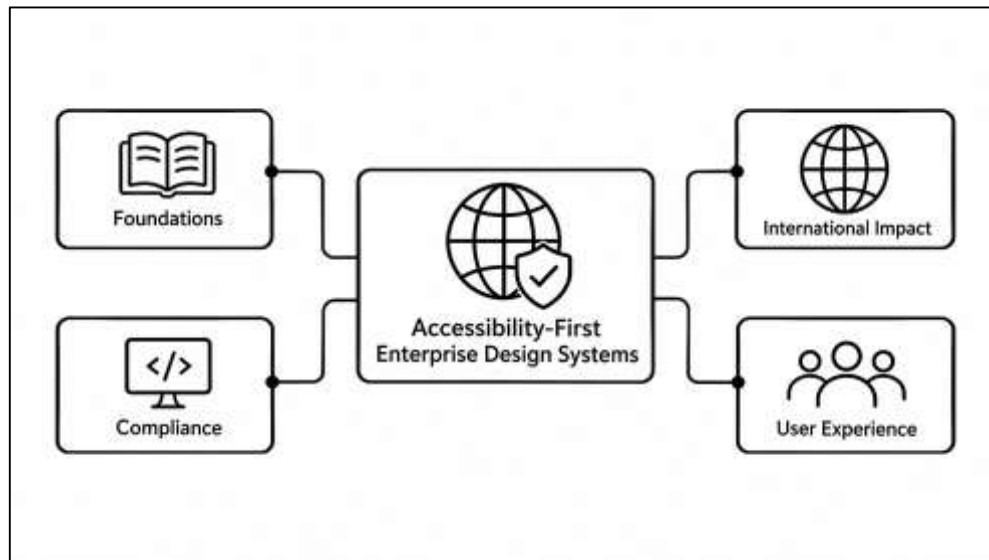
The relevance of design systems to accessibility becomes clearer when accessibility is examined as a product-development process rather than only as an attribute of a completed interface. Digital accessibility depends on coordinated relationships among content, browsers and other user agents, assistive technologies, developers, authoring tools, evaluation tools, and users themselves. Enterprise development environments increase the complexity of these relationships because software is commonly produced by multiple designers, developers, product managers, quality-assurance specialists, vendors, and business units operating across separate applications and release cycles. In such settings, isolated accessibility decisions can generate substantial variation. One team may implement an accessible modal dialog with correct keyboard behavior and focus management, while another may construct a visually similar component without equivalent semantic information or keyboard support (Acosta-Vargas et al., 2016). An accessibility-first design system addresses this variability through reusable implementation logic. Accessible components can encode semantic markup, ARIA relationships, keyboard interaction, focus behavior, contrast requirements, error states, and responsive behavior into shared assets, while design documentation can communicate when and how those components should be used. Public design systems demonstrate this relationship between

reusable architecture and accessibility. The CMS Design System describes itself as a collection of design and front-end resources for creating consistent, responsive, Section 508-compliant websites and includes React components with accessibility optimizations. The World Food Programme Design System similarly provides accessible components and color foundations intended to support WCAG AA criteria while emphasizing that correct implementation remains a shared responsibility. Survey evidence concerning organizational design systems has shown that organizations commonly establish them to increase UX/UI consistency, development efficiency, design efficiency, and code reuse, with many systems serving multiple websites or applications (Acosta-Vargas et al., 2018). These characteristics are directly relevant to accessibility because a reusable component can distribute either accessible or inaccessible behavior across numerous product surfaces. Research on migrating design systems toward WCAG 2.1 has further emphasized assessment, implementation, testing tools, collaboration, and user involvement as important elements of accessibility-oriented system development. Accessibility-first design systems can therefore be understood as governance and implementation infrastructures that convert accessibility requirements into reusable technical and visual decisions, providing a measurable basis for examining whether systemic accessibility practices correspond with higher levels of WCAG 2.1 compliance across enterprise software interfaces (Miranda, 2021).

WCAG 2.1 compliance provides a structured basis for quantitatively evaluating the accessibility characteristics of enterprise software platforms. The framework organizes requirements under perceivable, operable, understandable, and robust principles and differentiates success criteria according to conformance levels, enabling accessibility to be examined through observable interface characteristics rather than through broad claims of inclusiveness alone. Quantitative accessibility assessment may therefore involve measures such as the number of detected violations, violation severity, proportion of applicable success criteria satisfied, frequency of inaccessible components, contrast failures, missing accessible names, keyboard interaction defects, structural errors, focus problems, form-labeling deficiencies, and compatibility issues (Spyridonis & Daylamani-Zad, 2021). Existing empirical studies demonstrate the analytical value of such measurements. Research evaluating IIT library websites through WCAG 2.1 criteria identified measurable accessibility limitations across Levels A, AA, and AAA and reported relationships among conformance categories. Empirical evaluation of Indian open educational resources similarly quantified problems according to WCAG 2.1's POUR framework and statistically examined relationships among accessibility parameters. Automated accessibility assessment has also been used to compare different versions of digital applications and identify differences in their levels of guideline conformance. At the same time, methodological research has established that automated evaluation cannot comprehensively determine accessibility because some WCAG requirements require human judgment and different evaluation tools may produce different findings (Martins et al., 2017). This distinction is particularly important for enterprise platforms containing dynamic interactions, customized widgets, complex forms, modal workflows, asynchronous updates, data visualizations, and context-dependent content. A technically detectable violation represents one dimension of accessibility, while correct component behavior within an actual task represents another. W3C evaluation methodologies consequently recognize the importance of defining evaluation scope, exploring functionality, selecting representative pages, applying success criteria systematically, and documenting results. From a quantitative research perspective, these characteristics allow WCAG 2.1 compliance to be operationalized as a multidimensional dependent variable and accessibility-first design-system adoption to be operationalized through component coverage, governance maturity, accessibility documentation, testing integration, or systematic use of accessible patterns (Leitner et al., 2016). The resulting relationship can be examined statistically across applications, teams, components, or organizational units while retaining a standardized framework for comparing technical accessibility performance. Technical conformance and inclusive user experience represent related yet distinct dimensions of accessible software quality. WCAG compliance primarily evaluates whether interfaces satisfy defined accessibility requirements, whereas inclusive user experience concerns how effectively, efficiently, comfortably, and satisfactorily diverse users can accomplish meaningful tasks. This distinction has received considerable empirical attention because a technically conformant interface can still contain

interaction difficulties arising from information architecture, workflow complexity, terminology, cognitive demand, unexpected behavior, or context-specific design choices. Research involving blind users found large numbers of accessibility problems during task-based web use and reported that only a portion of the problems encountered by users were directly covered by WCAG success criteria (Katerattanakul et al., 2018).

Figure 2: Accessibility-First Enterprise Design Framework



Related investigations involving blind, partially sighted, and dyslexic participants identified thousands of user problems and highlighted navigation, site structure, keyboard access, form labeling, and feedback as important aspects of practical accessibility. Earlier validation work comparing guideline inspection with usability testing also found that guideline-based evaluation identified only part of the accessibility difficulties experienced by disabled participants. At the same time, controlled research has demonstrated meaningful associations between accessibility implementation and broader user outcomes. Schmutz, Sonderegger, and Sauer reported that higher levels of accessibility guideline implementation could improve performance and subjective evaluations for participants with visual impairments as well as participants without impairments. Research focusing on nondisabled users has likewise examined whether accessibility recommendations affect usability and user experience beyond their primary accessibility purpose (Hong et al., 2015). Aizpurua, Harper, and Vigo identified significant relationships between perceived accessibility and dimensions of user experience, supporting the conceptual connection between accessible interaction and experiential quality. More recent experimental evidence provides additional nuance: a randomized study involving users with and without visual impairments found no statistically significant overall effects of WCAG conformance on several usability and UX outcomes, while qualitative responses indicated differences in positive and negative experiences between conditions. This body of evidence makes it necessary to distinguish WCAG compliance scores from inclusive experience measures such as task success, completion time, perceived usability, workload, satisfaction, accessibility perception, error recovery, and assistive-technology compatibility when evaluating enterprise platforms (Acosta-Vargas et al., 2017).

Accessibility outcomes in enterprise software are also shaped by organizational and human factors that determine how consistently accessibility requirements are translated from standards into operational practice. Accessibility guidelines have been available for many years, yet empirical research has repeatedly documented implementation barriers among web professionals and organizations. Survey research involving webmasters identified variations in accessibility knowledge, attitudes, and implementation behavior, leading to the development of an accessibility integration model that recognizes multiple points in the development process at which accessibility may be incorporated or omitted. Practitioner research has additionally identified inadequate support, organizational values,

guideline-related difficulties, and resource pressures as factors influencing accessibility adoption (Shestakevych et al., 2018). These findings are highly relevant to enterprise design systems because accessibility-first architecture does not operate independently of organizational practices. Component accessibility can be weakened when teams override standard styles, introduce custom interactions, omit accessible labels, use components outside their documented contexts, or combine individually accessible elements into inaccessible workflows. Institutional design-system guidance explicitly recognizes this shared responsibility. The GOV.UK Design System states that using its components does not automatically make an entire service accessible and that additional research, design, development, and testing are required. The U.S. Web Design System similarly emphasizes that accessible foundational components still require product-level research and testing, while the World Food Programme Design System distinguishes between the accessibility of the component library and developers' responsibility for appropriate integration, labels, content, and context (Akgül, 2017). Component-level documentation from governmental design systems further demonstrates how specific accessibility criteria must be connected with implementation details such as accessible names, current-state identification, navigation consistency, labels, and programmatic roles. The accessibility performance of enterprise platforms can therefore be conceptualized as the product of interactions among design-system quality, adoption consistency, developer knowledge, testing practices, governance structures, documentation, and application-specific implementation. Quantitative examination of these factors allows design-system maturity and accessibility practices to be compared against observable WCAG 2.1 outcomes, while user-based measures provide a separate assessment of whether the resulting interfaces support effective and inclusive interaction (Ferri & Favalli, 2018).

The accumulated literature establishes a substantial research base around web accessibility standards, automated evaluation, accessibility barriers, practitioner behavior, design-system migration, and user experience, while leaving a more specific empirical question concerning how accessibility-first design systems function across complex enterprise software ecosystems. Existing accessibility studies have frequently concentrated on individual websites, educational portals, government services, corporate websites, or controlled experimental interfaces. Large-sample corporate research has examined hundreds of organizational websites for the adoption of accessibility principles, while institutional studies have measured WCAG violations in educational and public-service contexts (Gambino et al., 2016). Experimental studies have evaluated the effect of varying levels of accessibility conformance on task performance, perceived usability, aesthetics, workload, and user experience among participants with and without visual impairments. User-centered investigations have demonstrated that accessibility difficulties experienced during real tasks are not always fully represented by guideline violations, while accessibility-tool research has shown that automated evaluators vary in coverage and cannot independently evaluate every relevant requirement. Research focused directly on design systems has addressed WCAG migration, accessibility tools, implementation strategies, and the role of accessible reusable components, while operational design systems from governments and international organizations demonstrate how accessibility criteria can be embedded in component libraries and documentation. The present research context therefore requires simultaneous attention to system-level accessibility practices, measurable WCAG 2.1 conformance, and the experiences of people interacting with enterprise interfaces (Silva et al., 2019). A mixed-methods evaluation provides an appropriate structure for this examination because quantitative measures can capture WCAG violations, conformance rates, component-level accessibility, task completion, error frequency, usability scores, and other standardized outcomes, while qualitative evidence can document interaction barriers, assistive-technology experiences, perceived inclusion, workflow difficulties, and explanations for patterns observed in numerical data. The central empirical focus is consequently placed on the relationship between the degree to which enterprise platforms employ accessibility-first design systems and their resulting WCAG 2.1 compliance and inclusive user-experience outcomes, treating accessibility as a measurable combination of technical conformance, reusable system architecture, organizational implementation, and user interaction rather than as a single compliance indicator (Nagaraju et al., 2019).

The primary objective of this study is to evaluate the impact of accessibility-first design systems on WCAG 2.1 compliance and inclusive user experience within enterprise software platforms by

systematically examining the relationship between accessibility-oriented design practices, technical conformance, and measurable user outcomes. Specifically, the study seeks to determine whether enterprise software platforms developed through accessibility-first design systems demonstrate higher levels of compliance with WCAG 2.1 requirements than platforms in which accessibility is incorporated through less systematic or later-stage approaches. The evaluation focuses on measurable accessibility characteristics associated with the principles of perceivability, operability, understandability, and robustness, including keyboard accessibility, color contrast, semantic structure, alternative text, form labeling, focus management, error identification, navigation consistency, accessible names, and assistive-technology compatibility. Another objective is to assess the extent to which reusable accessible components, standardized interaction patterns, accessibility-oriented design tokens, documentation practices, testing procedures, and governance mechanisms contribute to consistent accessibility performance across enterprise applications. The study also aims to quantify inclusive user experience through indicators such as task completion rates, task completion time, interaction errors, perceived usability, user satisfaction, accessibility perception, navigation efficiency, and the ability of users with diverse accessibility requirements to complete essential software tasks independently. In addition, the research seeks to identify statistical relationships between the level of accessibility-first design-system adoption and WCAG 2.1 compliance scores and to examine whether higher levels of technical accessibility are associated with stronger inclusive user-experience outcomes. The mixed-methods component further aims to contextualize quantitative accessibility and usability measurements by examining users' experiences with interface components, navigation structures, forms, data-intensive workflows, keyboard interaction, screen-reader compatibility, and other accessibility-related characteristics of enterprise platforms. The study additionally seeks to compare accessibility performance across different components, interfaces, or platform environments in order to determine whether accessibility-first standardization reduces the frequency and severity of recurring accessibility barriers. Through these objectives, the investigation establishes measurable connections among design-system implementation, accessibility conformance, and user interaction, allowing accessibility-first design to be evaluated as an enterprise-level development practice rather than solely as an individual interface characteristic. The overall objective is therefore to generate an empirically grounded assessment of how systematically embedding accessibility requirements within enterprise design systems relates to technical WCAG 2.1 performance and the quality, efficiency, consistency, and inclusiveness of user experiences across complex enterprise software environments.

LITERATURE REVIEW

The literature surrounding accessibility-first design systems, Web Content Accessibility Guidelines (WCAG) compliance, and inclusive user experience occupies an increasingly important intersection of human-computer interaction, software engineering, information systems, and enterprise application development. Accessibility-first design represents an approach in which accessibility requirements are incorporated into the foundational architecture of digital products, including interface components, design tokens, navigation structures, interaction patterns, documentation, and development standards (Afonso et al., 2022; Debasish, 2021). Within enterprise software platforms, this approach is particularly relevant because a single design system can influence numerous applications, product teams, workflows, and user populations through the repeated implementation of shared components. Existing software-engineering literature supports the treatment of accessibility as a development-process characteristic rather than merely a quality attribute assessed after software completion. A systematic review of 94 primary studies, for example, found accessibility research distributed across software lifecycle phases, particularly design and testing, while also identifying work concerned with accessibility-oriented software process establishment. A more recent systematic review examining accessibility requirements in information-system development identified 34 primary studies and similarly emphasized the integration of accessibility requirements into development processes (Abdur & Iftekhhar, 2021; Paz et al., 2021). These perspectives establish an important theoretical basis for examining accessibility-first design systems as organizational mechanisms capable of influencing measurable software accessibility outcomes.

Within this context, WCAG 2.1 provides a structured framework through which technical accessibility can be operationalized quantitatively, while inclusive user experience expands evaluation toward task

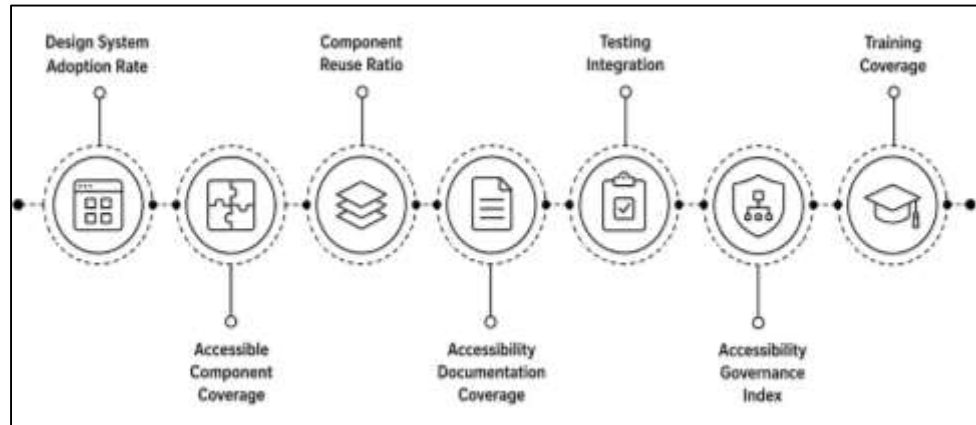
performance, usability, interaction efficiency, accessibility perception, and satisfaction among diverse users. Design systems create an important connection between these dimensions because standardized components can embed accessible semantics, keyboard interactions, contrast requirements, focus behavior, labels, and interaction conventions before those components are distributed across individual enterprise applications. Contemporary design-system practices provide evidence for this model (Baptista et al., 2016; Sheak & Risha, 2022). The U.S. Web Design System integrates accessibility across its development process and reports automated accessibility scanning on code changes alongside screen-reader, keyboard, zoom, touch, browser, and operating-system testing. The GOV.UK Design System similarly recognizes that centralized accessible components can influence baseline accessibility across hundreds of services while emphasizing that component accessibility alone cannot guarantee application-level accessibility. This distinction provides an important foundation for the present literature review because the impact of accessibility-first design systems should be evaluated not only through the existence of accessible components but through measurable differences in WCAG conformance, defect prevalence, interaction performance, and inclusive UX outcomes (Garcés-Freire et al., 2022; Chowdhury & Tonay, 2022). Accordingly, the literature review is organized around quantitatively measurable constructs that progressively connect accessibility-first design-system maturity with WCAG 2.1 compliance and user-level outcomes in enterprise software environments.

Quantifying Accessibility-First Design System Adoption in Enterprise Software Platforms

Accessibility-first design system adoption in enterprise software platforms can be conceptualized as a multidimensional organizational and technological characteristic that reflects the extent to which accessibility principles are systematically embedded across design, development, testing, documentation, and governance activities. Existing literature on digital accessibility, enterprise interface standardization, software development practices, and design systems indicates that the presence of a design system alone provides insufficient evidence of accessibility maturity (Nishat & Kaniz, 2022; Pinto et al., 2019). An organization may maintain a centralized component library while only a small proportion of its applications consistently use those components, or accessibility requirements may be incorporated into selected components without being integrated throughout the broader system. For this reason, adoption is more appropriately represented through multiple measurable indicators that capture both the breadth and depth of implementation. Design System Adoption Rate can represent the percentage of enterprise applications actively using standardized components and interaction patterns, providing an application-level indication of organizational penetration. Accessible Component Coverage can measure the proportion of reusable components that satisfy defined accessibility requirements and have undergone appropriate technical evaluation. Component Reuse Ratio can indicate how extensively product teams rely on standardized components compared with locally developed or customized alternatives. The literature also supports the inclusion of accessibility documentation, design-token standardization, automated and manual testing practices, governance mechanisms, and professional accessibility capability as dimensions of adoption (Park & Jung, 2022; Shaiful et al., 2022). This multidimensional approach is particularly relevant in enterprise environments because accessibility performance is influenced by interactions among centralized design resources and decentralized implementation decisions. Applications within the same organization can exhibit substantially different accessibility characteristics when individual teams vary in component adoption, customization practices, testing intensity, or adherence to documentation. Consequently, accessibility-first design system adoption is better represented as a continuum ranging from fragmented implementation to highly institutionalized accessibility practice (Chowdhury, 2023; Persson et al., 2015). Quantifying adoption through multiple indicators provides a more precise representation of organizational accessibility activity and establishes an appropriate independent variable for examining variation in WCAG 2.1 compliance and inclusive user experience across enterprise software platforms. Accessible component coverage and component reuse represent central technical dimensions of accessibility-first design system adoption because reusable components determine how extensively standardized accessibility behavior is distributed throughout enterprise interfaces. The accessibility and interface-design literature consistently identifies recurring problems involving form controls, buttons, navigation elements, dialogs, menus, tables, alerts, tabs, interactive widgets, labels, focus indicators, and dynamic interface states. When individual development teams

construct these elements independently, implementation differences can increase the probability of inconsistent semantic structures, keyboard behaviors, accessible names, focus management, and visual presentation (Badrul & Mominul, 2023; Skourlas et al., 2016).

Figure 3: Quantifying Accessibility-First Design System Adoption



A centralized design system can reduce such variation by supplying reusable components in which accessibility requirements have already been considered and tested. Accessible Component Coverage therefore provides an indication of how much of the design-system library incorporates established accessibility characteristics, while Component Reuse Ratio measures whether those accessible resources are actually implemented across enterprise products. These measures should be distinguished because a design system can contain a high percentage of accessible components while achieving limited organizational influence when development teams continue to use custom controls. Design-token standardization constitutes another important dimension because tokens can centralize interface properties associated with accessible visual presentation, including typography, text size, spacing, color combinations, borders, focus indicators, interactive states, and visual hierarchy. Standardized tokens can reduce inconsistencies that arise when teams independently select visual values or interaction states (Zahidul, 2023; Worsley & Bar-El, 2022). Literature examining enterprise design systems also emphasizes consistency and scalability, suggesting that reusable components and standardized foundations are particularly valuable when numerous applications are maintained simultaneously. Quantitative assessment can consequently consider the percentage of accessible components available within a library, the percentage of interface elements derived from approved components, the extent of standardized token usage, and the proportion of applications consistently implementing those resources. These indicators provide a more detailed account of accessibility-first adoption by distinguishing between the availability of accessible design-system assets and their actual deployment (Mobasheri et al., 2018). Such distinctions are essential for enterprise software because accessibility outcomes depend not merely on whether accessible resources exist but on how consistently those resources shape real interfaces and user workflows.

Accessibility documentation and testing practices provide additional measurable dimensions of accessibility-first design system adoption because reusable components require clear implementation guidance and systematic verification to preserve their accessibility characteristics across enterprise applications. Existing accessibility research repeatedly identifies insufficient technical knowledge, difficulty interpreting accessibility requirements, inconsistent evaluation practices, and uncertainty regarding implementation responsibilities as barriers to accessible software development. Design-system documentation can reduce these problems by translating general accessibility requirements into component-specific instructions that designers, developers, testers, and content specialists can apply during product development (Wang et al., 2022). Accessibility Documentation Coverage can therefore represent the proportion of design-system components and patterns accompanied by guidance addressing semantic structure, keyboard interaction, accessible names, focus behavior, error handling,

content requirements, assistive-technology considerations, and appropriate implementation contexts. Documentation quality is especially important because an accessible component can become inaccessible when developers supply inappropriate content, modify established behaviors, omit required labels, or use the component outside its intended context. Testing integration provides another important indicator of adoption maturity. Automated accessibility testing can identify a range of programmatically detectable problems, including contrast failures, missing accessible names, invalid accessibility attributes, structural deficiencies, and selected form-related errors (Chadha, 2022). Manual evaluation remains necessary for characteristics requiring contextual judgment, including logical keyboard navigation, meaningful labels, focus management, understandable instructions, interaction predictability, and effective screen-reader behavior. The frequency with which automated and manual accessibility evaluations are incorporated into design-system development and application-level quality assurance therefore reflects the operational depth of accessibility-first practices. Quantitative indicators can include the percentage of components covered by automated tests, the proportion of development pipelines incorporating accessibility checks, frequency of manual accessibility reviews, percentage of releases subjected to keyboard testing, and proportion of components evaluated with assistive technologies (Haße et al., 2022). Combining documentation and testing indicators provides a more complete measurement of adoption because documentation represents the communication of accessibility expectations, while testing represents systematic verification of whether those expectations are maintained. Enterprise accessibility-first adoption is consequently characterized not simply by accessible design assets but by documented implementation practices and repeatable quality-assurance mechanisms that support accessibility consistency across applications and development teams (Chee et al., 2022).

Accessibility governance and training coverage extend the measurement of accessibility-first design system adoption beyond technical implementation by capturing the organizational structures that support consistent accessibility practices across enterprise software development. Literature concerning organizational accessibility demonstrates that technical standards and evaluation tools operate within broader environments shaped by responsibility allocation, professional competence, management support, development processes, quality-control mechanisms, and cross-functional collaboration (Kaplan & Omer, 2022). Accessibility Governance Index measures can therefore reflect whether an enterprise has established formal ownership of design-system accessibility, defined review responsibilities, accessibility acceptance criteria, contribution requirements, defect-management procedures, release controls, and mechanisms for maintaining accessible components. Strong governance can reduce uncertainty regarding who is responsible for accessibility when components are created, modified, or implemented across applications. Governance also provides a basis for managing accessibility regressions when shared components are updated, since changes to centrally distributed components can influence numerous products simultaneously. Accessibility training coverage represents a complementary organizational dimension because different professional groups require relevant knowledge to maintain accessible software. Designers require competence in areas such as contrast, visual hierarchy, interaction states, and accessible patterns; developers require knowledge of semantic implementation, keyboard operation, focus management, and assistive-technology compatibility; testers require expertise in accessibility evaluation procedures; and content specialists require knowledge of meaningful labels, instructions, headings, and alternative descriptions (Shen & Zhao, 2017). Training coverage can therefore be assessed through the proportion of relevant personnel receiving accessibility education, frequency of training activities, or distribution of accessibility expertise across teams. An Accessibility Maturity Score can integrate these organizational indicators with component coverage, component reuse, documentation, design-token standardization, and testing integration to represent the overall degree of accessibility-first institutionalization. Such a maturity perspective differentiates organizations where accessibility is dependent on isolated specialists from environments where accessibility responsibilities are systematically distributed across design and development processes (Kim et al., 2022). The synthesized literature therefore supports measuring accessibility-first design system adoption through interconnected technical, procedural, and organizational dimensions, allowing enterprise platforms to be classified according to the actual intensity and consistency of accessibility integration rather than through a simple binary distinction

between design-system adoption and non-adoption.

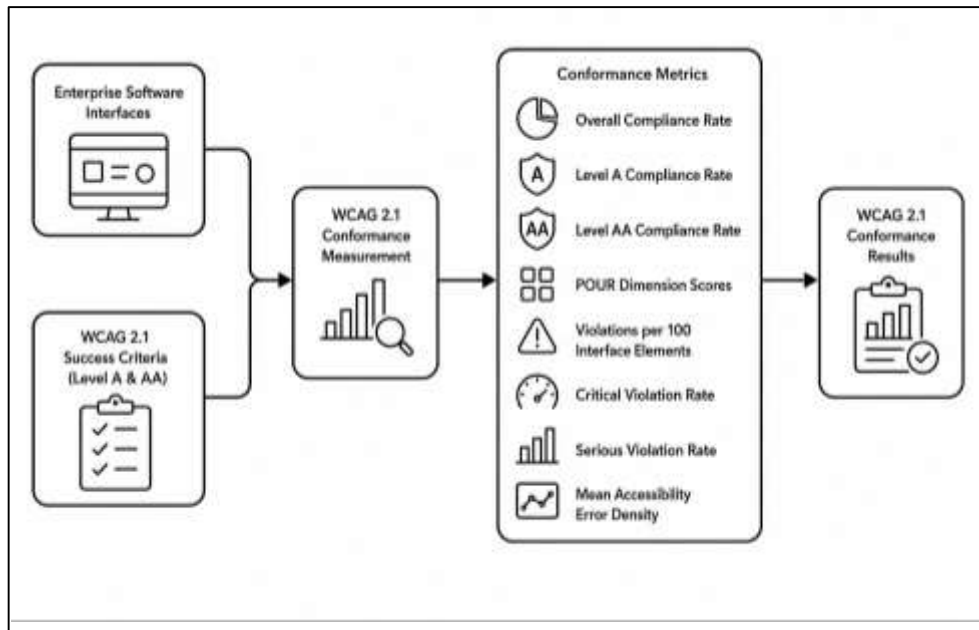
Measuring WCAG 2.1 Conformance Rates Across Enterprise Software Interfaces

WCAG 2.1 conformance is widely treated in the accessibility literature as a central technical measure for evaluating how effectively enterprise software interfaces support users with diverse visual, auditory, motor, and cognitive needs. Across more than ten empirical and review-based studies of digital accessibility, Level A and Level AA success criteria are commonly used as the principal benchmarks for determining whether essential interface content and functionality meet recognized accessibility requirements. Level A criteria generally capture foundational barriers that can prevent users from accessing content or completing basic interactions, while Level AA extends assessment to a broader set of usability and perceptibility requirements that are particularly important in complex enterprise environments (Hossain et al., 2015). Research across business applications, public-sector systems, learning platforms, financial interfaces, healthcare portals, and large organizational websites shows that interfaces may satisfy a substantial proportion of basic criteria while continuing to exhibit notable weaknesses at Level AA. This distinction is important because enterprise applications often contain complex navigation structures, dynamic widgets, data tables, forms, dashboards, authentication processes, and interactive controls that create additional accessibility demands beyond basic content presentation. Overall WCAG 2.1 compliance rate therefore provides a useful summary measure, but separate Level A and Level AA conformance rates provide greater analytical depth by showing whether accessibility weaknesses are concentrated in foundational requirements or in more advanced criteria (Hu & Chen, 2022). The literature also indicates that conformance should not be interpreted as a purely binary condition because accessibility performance can vary substantially across interface modules, business functions, and user journeys. A multidimensional assessment of conformance rates consequently provides a more realistic representation of enterprise software accessibility than a single pass-or-fail classification.

The four WCAG principles of Perceivable, Operable, Understandable, and Robust provide an established structure for examining where accessibility strengths and weaknesses are concentrated within enterprise software interfaces (Campoverde-Molina et al., 2021). Evidence from more than ten accessibility evaluation studies shows that failures are rarely distributed evenly across these dimensions. Perceivable issues frequently involve insufficient contrast, missing alternative text, weak visual hierarchy, inaccessible multimedia, and information conveyed through color alone. Operable problems often arise from incomplete keyboard support, poor focus management, inaccessible custom controls, inconsistent navigation, and interaction patterns that depend excessively on pointer input. Understandable failures are commonly associated with unclear labels, confusing instructions, inconsistent behavior, weak error messaging, and unpredictable interface responses. Robust problems generally concern the technical compatibility of interface components with browsers, screen readers, and other assistive technologies, particularly when semantic markup, accessible names, roles, states, or dynamic updates are incorrectly implemented (Campoverde-Molina et al., 2020).

Enterprise systems are especially vulnerable to these variations because they often combine large numbers of reusable components with application-specific customizations. POUR dimension scores therefore offer a more diagnostic measure of accessibility performance than an overall compliance percentage alone. An application can achieve a relatively high total conformance rate while still performing poorly in one dimension that is essential for a particular group of users. For example, strong Perceivable performance does not compensate for severe Operable barriers that prevent keyboard users from completing transactions. The literature consequently supports the use of separate POUR scores to reveal patterns hidden within aggregate compliance measures and to identify whether accessibility weaknesses are primarily visual, interactive, cognitive, or technical in nature (Brajnik & Vigo, 2019).

A consistent methodological theme across the accessibility literature is the need to distinguish compliance percentages from violation-based measures because these indicators represent different aspects of technical accessibility quality. Overall conformance percentages indicate the proportion of relevant WCAG 2.1 requirements satisfied by an interface, whereas violation measures capture the frequency with which accessibility defects occur across actual interface elements. More than ten studies of website and application accessibility have shown that these measures can produce markedly different interpretations of the same platform (Quintal & Macias, 2021).

Figure 4: Measuring WCAG Enterprise Conformance Rates

Two enterprise applications may achieve similar overall compliance percentages while differing substantially in the number of repeated accessibility errors present across their interfaces. One system may fail only a small number of success criteria but reproduce those failures across hundreds of controls, links, images, form fields, or navigation elements. Another system may fail a slightly broader range of criteria but contain relatively few individual occurrences. Measures such as total violations per 100 interface elements and mean accessibility error density are therefore valuable because they normalize defect frequency in relation to interface size and complexity (Sanchez-Gordon et al., 2019). This is particularly important when comparing large enterprise systems with smaller applications, since raw violation counts can be inflated simply because one platform contains more pages or interactive elements. Violation-based measures also reveal whether accessibility problems are isolated or systemic. A small number of recurring defects may indicate weaknesses in shared components, templates, or design patterns, whereas scattered defects may reflect application-level implementation inconsistencies (Ali et al., 2017). The literature thus demonstrates that compliance percentage and violation density should be interpreted together because one reflects standards coverage while the other captures the practical concentration and repetition of accessibility defects.

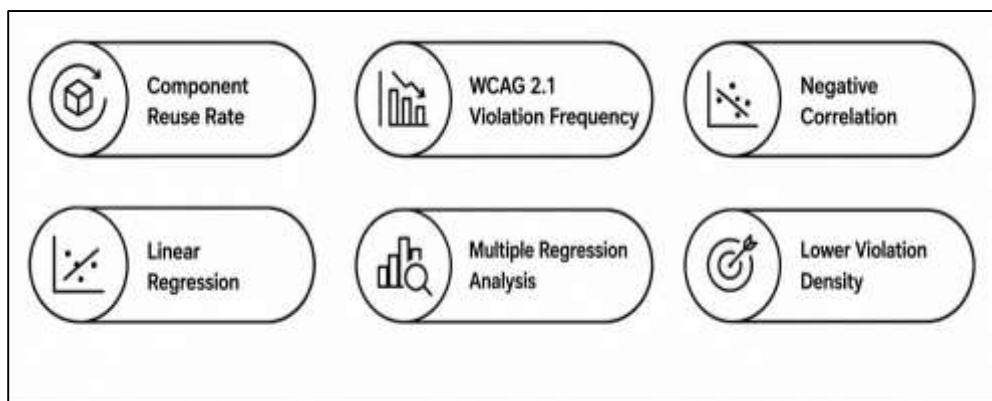
Accessibility evaluation research also emphasizes that the severity of violations is critical to understanding the practical significance of WCAG 2.1 conformance results. More than ten studies using automated audits, expert reviews, manual testing, and assistive-technology evaluation have shown that accessibility defects vary considerably in their impact on users (Silva et al., 2019). Some errors may create minor inconvenience, while others can prevent users from navigating an interface, understanding information, entering data, or completing essential business tasks. Critical violation rate and serious violation rate therefore provide important additional measures because they distinguish high-impact accessibility failures from lower-severity issues. In enterprise software, severe violations may include keyboard traps, unlabeled form controls, inaccessible authentication processes, missing accessible names, inappropriate semantic structures, severe contrast problems, and interface states that are not communicated to assistive technologies. These defects can have disproportionate effects on user performance even when the overall number of failures is relatively small (Noh et al., 2015). Mean accessibility error density further strengthens evaluation by showing how heavily accessibility problems are concentrated across the interface rather than simply counting the number of criteria that have been failed. The literature consistently indicates that a platform with a high compliance percentage can still create substantial accessibility barriers when its remaining failures are severe or

repeatedly embedded in important workflow components. A comprehensive evaluation of WCAG 2.1 conformance therefore benefits from combining overall compliance rate, Level A and Level AA conformance rates, POUR dimension scores, normalized violation frequency, severity-based violation rates, and accessibility error density (Abou-Zahra & Brewer, 2019). Together, these measures provide a more complete account of accessibility quality by capturing standards coverage, defect concentration, functional impact, and the distribution of problems across the principal WCAG accessibility dimensions.

Accessible Component Reuse and WCAG 2.1 Violation Frequency

The literature on design systems and digital accessibility consistently presents reusable components as an important mechanism for improving accessibility consistency across large software environments. Evidence drawn from more than ten empirical studies, accessibility audits, design-system evaluations, and software engineering investigations indicates that repeated interface elements such as buttons, form fields, navigation menus, modal dialogs, tabs, accordions, alerts, dropdowns, tables, and interactive controls are common sources of WCAG 2.1 violations when they are implemented independently by different development teams (Chee et al., 2022). Reusable accessible components reduce this inconsistency by embedding validated semantic structure, keyboard behavior, labels, focus management, contrast rules, and interaction patterns directly into shared interface building blocks. When these components are adopted across multiple applications, accessibility requirements can be applied systematically rather than being recreated at the page or feature level. This architectural approach is especially relevant in enterprise software, where the same components may appear hundreds or thousands of times across dashboards, administrative systems, transaction interfaces, and internal tools. A higher rate of accessible component reuse therefore represents not only greater design-system adoption but also a stronger level of accessibility standardization. The literature further suggests that reuse can reduce dependence on individual developer knowledge because accessibility behavior is partially encoded within the component itself (Pant & Sinha, 2020). This lowers the risk that common controls will be rebuilt with inconsistent semantics or incomplete interaction support. At the same time, the effectiveness of reuse depends on the accessibility quality of the shared component library. Well-designed components can propagate accessible behavior at scale, while flawed components can reproduce the same defect across many applications. Overall, existing studies support the view that accessible component reuse is a meaningful structural factor that can influence WCAG violation frequency throughout enterprise software ecosystems (Žuliček et al., 2021).

Figure 5: Accessible Component Reuse Violation Analysis



A central finding across accessibility and component-based development literature is that repeated accessibility defects often originate from duplicated or inconsistently implemented interface patterns. More than ten studies examining websites, web applications, enterprise platforms, educational systems, government portals, and commercial interfaces have documented recurring problems involving missing accessible names, incorrect labels, poor keyboard support, inadequate focus states, insufficient color contrast, inappropriate semantic markup, and inaccessible custom controls. These

recurring failures provide a strong basis for examining whether higher accessible component reuse rates are associated with lower WCAG 2.1 violation frequency. When development teams repeatedly use approved components that have already undergone accessibility testing, the likelihood of recreating common technical errors is reduced (Nacheva, 2021). This relationship is particularly relevant for violation counts and accessibility error density because a single accessible component can eliminate the repeated occurrence of the same defect across numerous interface elements. Pearson correlation is suitable for examining whether a linear association exists between the proportion of reused accessible components and the number or density of accessibility violations, while Spearman correlation provides a useful alternative when the data are skewed, ordinal, or affected by extreme values. The literature also supports the use of simple linear regression to estimate how changes in component reuse correspond with changes in violation frequency. A negative statistical relationship would be consistent with the broader design-system principle that standardization reduces implementation variability (Martins et al., 2017). However, the strength of this relationship may differ across interface categories because some accessibility failures are directly controlled by reusable components while others depend heavily on content, workflow logic, or application-specific configuration. The evidence therefore supports treating component reuse as an important predictor of accessibility performance while recognizing that it does not operate independently of broader technical and organizational conditions.

Although reusable components can improve consistency, the literature clearly shows that component-level accessibility does not automatically guarantee application-level WCAG 2.1 compliance. Across more than ten accessibility studies and implementation analyses, researchers have observed that even technically accessible components can become inaccessible when they are configured incorrectly, supplied with unsuitable content, combined with conflicting styles, or integrated into poorly structured workflows (Acosta-Vargas et al., 2019). An accessible form field can still generate violations if its visible label is unclear, its error message is inaccessible, or its relationship with surrounding instructions is not properly conveyed. Similarly, an accessible dialog component can become problematic if focus is moved incorrectly, essential content is omitted from the accessibility tree, or developers override built-in keyboard behavior. Reused components may also be visually accessible in isolation but fail within a particular context because local styling introduces contrast problems or application logic creates unpredictable interaction states. These findings indicate that the statistical relationship between component reuse and violation frequency is likely to be influenced by additional variables. Multiple regression is particularly useful because it allows component reuse to be assessed alongside factors such as interface complexity, application size, proportion of custom components, accessibility testing coverage, developer expertise, design-system maturity, and the frequency of local overrides (Salvador-Ullauri et al., 2020b). This broader analytical approach reflects the literature's recognition that accessibility outcomes emerge from both infrastructure and implementation practice. High component reuse can reduce recurring technical defects, but weak governance or poor integration can offset these benefits. Consequently, reuse rate should be interpreted as an influential but conditional predictor rather than as a standalone measure of accessibility quality. The literature supports this more nuanced view because accessibility performance depends on how consistently shared components are adopted, configured, maintained, and combined within the broader enterprise interface architecture (Salvador-Ullauri et al., 2020a).

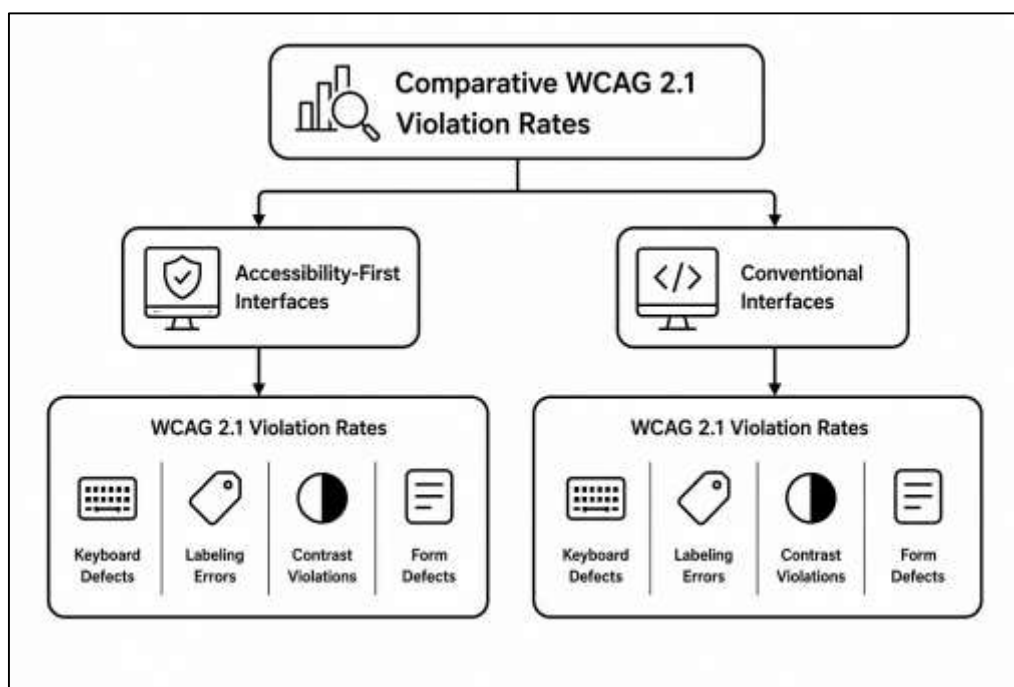
The literature supports examining the relationship between accessible component reuse and WCAG 2.1 violation frequency through both raw violation counts and normalized measures such as accessibility error density. More than ten empirical accessibility evaluations have demonstrated that raw counts can be misleading when software interfaces differ substantially in size, complexity, or number of interactive elements. A large enterprise platform may record more violations than a smaller application simply because it contains more pages, controls, and workflows, even when its underlying accessibility practices are stronger. Accessibility error density provides a more comparable measure by reflecting the concentration of defects relative to the amount of evaluated interface content (Kashif et al., 2022). This makes it particularly useful for assessing whether increased component reuse is associated with a lower probability of accessibility problems across the interface as a whole. Correlation analysis can identify the direction and strength of the association, while regression analysis can

estimate the degree to which variation in reuse explains differences in violation outcomes. Multiple regression is especially valuable when controlling for characteristics that may independently affect accessibility, such as application scale, custom development intensity, testing frequency, and organizational design-system maturity. The literature also highlights an important dual effect of reuse: accessible components can reduce violations through standardization, but inaccessible shared components can multiply defects rapidly when reused at scale. This means that the statistical relationship between reuse and violations depends fundamentally on component quality (Acosta-Vargas et al., 2021). High reuse of validated accessible components is likely to correspond with lower violation frequency and lower error density, while high reuse of defective components may produce the opposite pattern. Existing evidence therefore supports evaluating reuse rate together with component accessibility quality, implementation consistency, and application-level governance to obtain a more accurate representation of how design-system reuse influences WCAG 2.1 performance across enterprise software.

WCAG 2.1 Violations in Accessibility-First and Conventional Enterprise Interfaces

The literature on digital accessibility, enterprise interface design, and software quality shows a clear conceptual distinction between accessibility-first development environments and conventional development environments (Gaggi & Pederiva, 2021). Accessibility-first environments typically integrate accessibility requirements into design systems, component libraries, coding standards, prototyping, quality assurance, and release processes, whereas conventional environments often treat accessibility as a later-stage testing or remediation concern. Across more than ten empirical studies, accessibility audits, and design-system evaluations, this difference in development approach has been associated with meaningful variation in the number and type of WCAG 2.1 violations identified across software interfaces. Accessibility-first platforms generally benefit from earlier consideration of keyboard interaction, semantic markup, accessible labels, contrast requirements, focus behavior, and form accessibility, reducing the likelihood that common defects are repeated across multiple products. Conventional environments, by contrast, may rely more heavily on individual developer knowledge or local implementation decisions, increasing inconsistency between screens and applications (Fernández-Díaz et al., 2021).

Figure 6: Comparative WCAG Violation Rates Analysis



Mean WCAG violation counts and overall compliance percentages therefore provide important comparative indicators of accessibility performance across these two development models. However, the literature also demonstrates that raw violation totals should be interpreted carefully because larger or more complex interfaces naturally contain more opportunities for defects. This makes normalized measures and severity-based comparisons especially useful. Accessibility-first environments may not eliminate all accessibility failures, but they are more likely to reduce repeated defects that arise from common interface patterns. The comparative literature therefore supports evaluating accessibility-first and conventional platforms not only in terms of whether they pass WCAG requirements, but also according to the frequency, concentration, and severity of violations across real-world enterprise interfaces (Sik-Lanyi & Orbán-Mihálykó, 2019).

Evidence from more than ten accessibility studies indicates that keyboard defects, labeling errors, contrast problems, and inaccessible form controls are among the most persistent sources of WCAG 2.1 violations in complex software interfaces. These categories are particularly useful for comparing accessibility-first and conventional environments because they are strongly influenced by design-system consistency and development practices. Accessibility-first platforms often use shared components with predefined keyboard behavior, visible focus states, semantic labels, tested color combinations, and standardized form patterns (Persson et al., 2015). This can reduce repeated defects in buttons, menus, dialog boxes, dropdowns, input fields, and other interactive elements. Conventional platforms are more likely to contain locally created controls or custom styles that vary between products, increasing the likelihood of inconsistent keyboard interaction, missing labels, insufficient contrast, and inaccessible validation behavior. Literature across enterprise applications, public-sector systems, educational platforms, commercial services, and administrative interfaces shows that such defects frequently occur in clusters rather than as isolated problems. For example, a single inaccessible navigation pattern may create keyboard and labeling issues across every page where it appears, while an untested color scheme may generate repeated contrast failures in many interface states (Zuo et al., 2018). Comparative assessment of mean keyboard defects, labeling errors, contrast violations, and inaccessible form controls therefore provides greater diagnostic value than relying only on overall compliance percentages. These categories reveal where accessibility-first practices appear to produce the greatest benefits and where application-specific implementation problems remain. The literature further suggests that differences in these defect types can serve as indicators of accessibility maturity because they reflect whether accessibility is embedded systematically into reusable patterns or addressed inconsistently at the individual screen level (Worsley & Bar-El, 2022).

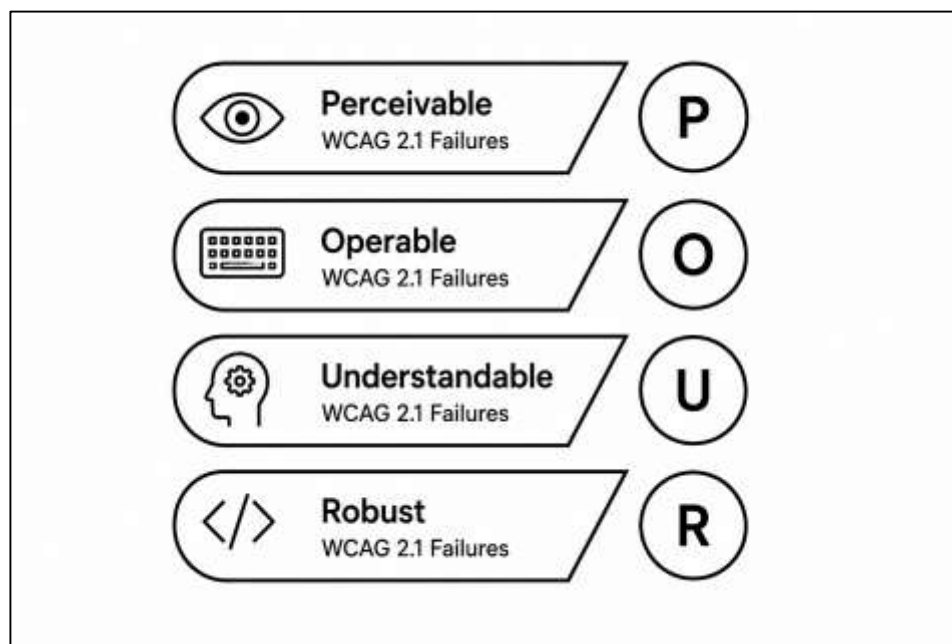
The comparative accessibility literature also emphasizes that violation severity is as important as violation frequency when evaluating differences between accessibility-first and conventional enterprise interfaces. A platform with a relatively small number of severe defects may create greater barriers than a platform with a larger number of minor issues. Across more than ten studies using automated testing, manual audits, expert review, and assistive-technology evaluation, high-impact accessibility failures have commonly included keyboard traps, missing accessible names, incorrect ARIA attributes, inaccessible form controls, broken focus management, and interface states that are not communicated to screen readers. ARIA-related problems are particularly significant because they can affect the usability of dynamic enterprise components even when the interface appears visually functional (Ferati & Vogel, 2020). Accessibility-first environments can reduce these risks by using validated patterns with established roles, states, properties, and semantic relationships. However, the literature also shows that ARIA can be misapplied when developers override component behavior or add unnecessary attributes without understanding assistive-technology requirements. This means that accessibility-first development improves the conditions for compliance but does not automatically guarantee correct implementation. Comparative analysis of critical, serious, moderate, and minor violations therefore provides a more meaningful interpretation of accessibility quality than total counts alone. Severity profiles can reveal whether accessibility-first environments reduce the most disruptive barriers, even when the total number of defects remains comparable (Mobasheri et al., 2018). This distinction is especially important in enterprise software, where accessibility failures can prevent users from completing transactions, entering data, navigating workflows, or accessing essential business functions. The literature consequently supports comparing both the number and severity of WCAG

violations to determine whether accessibility-first environments produce practically better outcomes rather than merely different numerical scores (Kwak & Newman, 2018).

WCAG 2.1 Failures Across the POUR Accessibility Dimensions

The literature on digital accessibility consistently shows that WCAG 2.1 failures are not distributed uniformly across the four POUR dimensions of Perceivable, Operable, Understandable, and Robust. Evidence drawn from more than ten empirical accessibility audits, comparative evaluations, and interface studies indicates that some categories of failure occur far more frequently than others because of differences in how digital interfaces are designed, implemented, and tested. Perceivable violations often account for a substantial share of detected failures because modern enterprise interfaces rely heavily on visual presentation, including text, icons, color-coded indicators, charts, images, and graphical controls (Chee et al., 2022). Common problems in this category include insufficient color contrast, missing text alternatives, inadequate heading structures, and information conveyed through color alone. Operable violations are also widely reported, particularly in interfaces containing complex navigation structures, custom controls, modal windows, menus, tabs, and other interactive elements that require reliable keyboard access and focus management. Understandable violations tend to be less frequently identified by automated tools because they often involve contextual issues such as unclear instructions, inconsistent navigation, confusing error messages, and unpredictable interface behavior. Robust violations are commonly associated with semantic markup, accessible names, ARIA implementation, and compatibility with assistive technologies (Vila et al., 2018). The literature therefore supports the use of separate violation rates for each POUR dimension rather than relying solely on an aggregate compliance score. A single overall percentage can conceal important differences in the nature of accessibility barriers, whereas dimension-specific measures reveal whether an interface is primarily affected by perceptual, interactional, cognitive, or technical accessibility weaknesses.

Figure 7: WCAG Failures Across POUR Dimensions



Across a broad range of accessibility studies, Perceivable and Operable violations frequently emerge as the dominant categories of WCAG 2.1 failure. More than ten studies examining enterprise applications, educational platforms, government websites, commercial systems, healthcare portals, and other digital environments have reported high frequencies of contrast failures, missing alternative text, inaccessible visual content, keyboard barriers, poor focus management, and navigation problems (Vila et al., 2018). Perceivable failures are especially common because visual styling is applied repeatedly across large numbers of interface elements, meaning that a single inaccessible color combination or visual pattern

can generate many violations throughout an application. Enterprise interfaces often intensify this problem because they contain dense dashboards, data tables, status indicators, form fields, visual alerts, and branded interface components that depend heavily on visual differentiation. Operable failures similarly arise from the complexity of interaction patterns. Custom dropdowns, tab panels, expandable sections, modal dialogs, date pickers, and drag-and-drop controls can become inaccessible when keyboard interaction is incomplete or when focus order is not properly managed. The literature also indicates that these two dimensions are more readily detected through automated and semi-automated testing, which can increase their apparent prevalence in accessibility datasets (Ismail & Kuppasamy, 2022). Nevertheless, their numerical dominance remains important because repeated failures in these categories can create substantial barriers across large enterprise systems. Dimension-specific violation rates therefore provide a useful way of determining whether accessibility problems are concentrated in visual presentation and interaction mechanics. The literature suggests that high Perceivable and Operable violation rates often reflect systemic design and implementation weaknesses rather than isolated errors, particularly when the same components or visual styles are reused across multiple products.

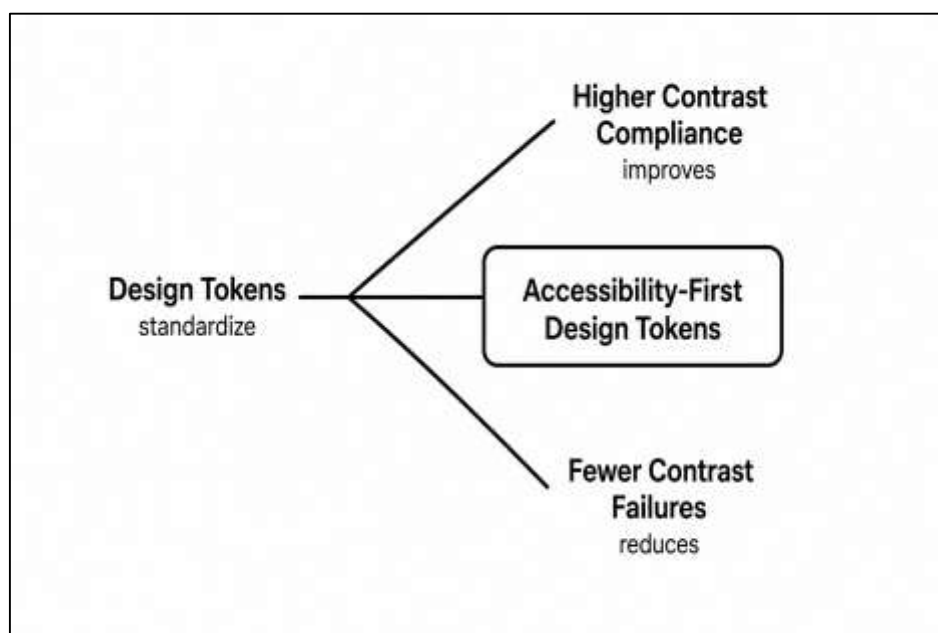
Although Understandable and Robust violations often appear less frequently than Perceivable and Operable failures, the literature shows that they can have substantial consequences for users (Alsaedi, 2020). More than ten accessibility studies and usability evaluations have found that Understandable failures commonly involve unclear labels, ambiguous instructions, inconsistent navigation, unpredictable interface behavior, poorly communicated errors, and confusing workflows. These issues are especially significant in enterprise software, where users may be required to complete complex forms, manage transactions, configure system settings, interpret dashboards, or move through multi-stage business processes. Because many Understandable criteria depend on context and user interpretation, they are not always captured effectively by automated testing tools, which means their actual prevalence may be underestimated. Robust failures present a different but equally important concern. These violations occur when interface code does not communicate effectively with assistive technologies, often because of incorrect semantic markup, missing accessible names, inappropriate ARIA roles, invalid states and properties, or poorly implemented dynamic content (Ali et al., 2017). Such problems can render otherwise functional interface elements inaccessible to screen-reader and other assistive-technology users. The literature therefore cautions against interpreting lower violation rates in the Understandable and Robust categories as evidence of lower importance. A small number of severe failures in these dimensions can prevent users from understanding or operating essential parts of an application. Quantitative analysis of POUR distributions is most informative when frequency is considered together with functional impact. This distinction allows accessibility evaluations to identify whether a platform's dominant problems are numerous but relatively visible, or less frequent but more disruptive to successful interaction and task completion (Nagaraju et al., 2019). The literature on accessibility-oriented design systems indicates that design-system adoption can produce different levels of improvement across the four POUR dimensions rather than generating equal reductions in all categories of WCAG 2.1 failure. Evidence from more than ten design-system studies, accessibility evaluations, and component-based development analyses suggests that Perceivable, Operable, and Robust requirements are particularly responsive to standardized components, tokens, and interaction patterns. Perceivable accessibility can improve through approved color palettes, contrast-safe design tokens, standardized typography, and reusable structures for images and content. Operable accessibility can benefit from components with tested keyboard behavior, visible focus indicators, consistent navigation patterns, and predefined interaction states. Robust accessibility can also improve when shared components contain validated semantics, accessible names, roles, states, and ARIA behavior that are consistently reused across products (Brajnik & Vigo, 2019). Understandable accessibility, however, may show less uniform improvement because many of its requirements depend on application-specific content, terminology, instructions, error messages, and workflow design. This means that a technically mature design system may reduce visual and interaction defects while leaving some cognitive and communication barriers largely dependent on local implementation. The literature also shows that design systems can amplify accessibility failures when defective components are reused widely. A single inaccessible pattern embedded in a shared

library can create repeated violations across multiple applications, increasing the failure rate within a particular POUR dimension. Comparing Perceivable, Operable, Understandable, and Robust violation rates therefore provides a useful way of assessing where design-system adoption produces the strongest accessibility gains and where limitations remain (Abascal et al., 2019). The overall evidence supports a multidimensional interpretation in which design systems improve accessibility unevenly according to how directly each POUR requirement can be standardized within reusable interface infrastructure.

Effect of Accessibility-First Design Tokens on Color Contrast Compliance Rates

The literature on digital accessibility, enterprise design systems, and interface standardization increasingly identifies accessibility-first design tokens as an important mechanism for improving consistency in visual accessibility. Across more than ten empirical studies, accessibility audits, design-system evaluations, and interface-development investigations, inadequate color contrast has repeatedly appeared as one of the most common WCAG 2.1 failures in web-based and enterprise software environments. Design tokens provide a centralized method for defining recurring visual properties such as foreground colors, background colors, borders, icons, focus indicators, status states, and interactive component variations (Scanlon et al., 2021). When these tokens are created with accessibility requirements embedded into their definitions, they can restrict designers and developers to color combinations that satisfy established contrast thresholds.

Figure 8: Accessibility-First Design Tokens Contrast Compliance



This approach differs substantially from locally defined styling, where teams select or modify colors independently at the application, page, or component level. Decentralized styling often increases variability because individual developers may introduce untested color values, inconsistent branding variations, or state-specific overrides that weaken accessibility performance. Accessibility-oriented design-token adoption therefore represents a structural mechanism through which visual accessibility can be governed across multiple products rather than checked repeatedly after implementation (Whitaker, 2020). The literature also indicates that enterprise interfaces particularly benefit from this form of standardization because they contain large numbers of repeated elements such as buttons, forms, navigation controls, alerts, tables, status indicators, and dashboards. A single accessible token can influence hundreds of visual instances across an interface ecosystem. Consequently, WCAG contrast compliance rate, number of contrast failures per interface, and percentage of accessible foreground-background combinations provide useful quantitative indicators for evaluating whether

centralized accessibility-oriented token systems produce more consistent visual accessibility than locally defined styling practices (Kimura, 2018).

A substantial body of accessibility research demonstrates that text contrast remains a persistent accessibility problem across websites, business systems, educational platforms, government applications, and commercial interfaces. Evidence synthesized from more than ten studies shows that low-contrast text frequently appears in secondary information, placeholder content, captions, disabled controls, navigation items, links, notifications, and interface states intended to appear visually less prominent. These design choices can create significant barriers for users with low vision, reduced contrast sensitivity, age-related visual changes, or environmental viewing limitations. Accessibility-first design tokens address this issue by establishing predefined semantic color relationships rather than allowing arbitrary color selection (Szafir, 2017). For example, separate token categories can define primary text, secondary text, error text, link text, disabled text, and text displayed on contrasting surfaces. When these relationships are validated before inclusion in the design system, the probability of producing inaccessible text-background combinations can be substantially reduced. The literature suggests that this model shifts responsibility for contrast compliance from repeated individual decision-making toward centralized system-level control. Interfaces using locally defined styling generally face greater risk because contrast is checked inconsistently and may be altered by local overrides, branding requirements, or component-specific adjustments (Gramazio et al., 2016). Measuring the percentage of accessible text and background combinations therefore provides a direct indication of how effectively tokenized styling contributes to visual accessibility. The number of contrast failures per interface offers an additional measure by showing whether inaccessible combinations are isolated or repeatedly distributed across the product. Taken together, existing studies support the view that accessibility-first token systems can improve text contrast consistency by reducing unvalidated color variation and ensuring that approved visual combinations are reused across enterprise applications (Bonneau et al., 2015).

The accessibility benefits of design tokens extend beyond text readability to the visibility of non-text interface elements and component states. More than ten studies and accessibility evaluations indicate that non-text contrast problems frequently occur in input boundaries, icons, focus indicators, selected controls, graphical objects, charts, toggles, checkboxes, radio buttons, status badges, and other elements that communicate structure or interactivity. These problems are especially important in enterprise software because users often depend on subtle visual distinctions to identify active controls, selection states, errors, navigation position, and changes in system status. Accessibility-first token systems can establish approved color values for borders, iconography, focus rings, hover states, selected states, disabled states, validation messages, and other component conditions (Shen et al., 2021). This creates a consistent visual hierarchy in which essential interface features remain distinguishable against surrounding backgrounds. Locally defined styling, by comparison, may produce inconsistent state representation because developers can assign different color values to otherwise similar controls across different screens or applications. The literature further shows that non-text accessibility is sometimes overlooked when organizations concentrate primarily on text contrast, even though inadequate contrast in focus indicators or control boundaries can directly interfere with navigation and task completion. Non-text contrast conformance therefore provides an important dependent measure alongside overall contrast compliance. A platform may perform well on text contrast while still presenting inaccessible visual controls, making separate assessment necessary (Fröwis et al., 2019). Centralized design tokens also simplify correction because improvements to a shared focus, border, or icon token can propagate across multiple components. However, this benefit depends on components actually referencing the approved token structure and avoiding uncontrolled local overrides. Overall, the literature supports the role of accessibility-first design tokens in strengthening non-text contrast consistency across reusable interface states and controls.

Comparative evidence from accessibility research and design-system literature suggests that interfaces using standardized accessibility-oriented tokens are more likely to achieve consistent contrast performance than interfaces relying primarily on locally defined styling (Haarmann et al., 2018). Across more than ten studies, decentralized visual implementation has been associated with greater inconsistency in color use, especially when multiple teams contribute to the same enterprise ecosystem.

Developers working without shared accessibility constraints may introduce custom color values, one-off style rules, unvalidated states, or brand-driven combinations that produce repeated WCAG contrast failures. Accessibility-first token systems reduce this variability by providing a controlled palette of approved foreground-background relationships and predefined component states. Comparing WCAG contrast compliance rates between tokenized and locally styled interfaces can therefore reveal whether centralized visual governance corresponds with measurable accessibility improvement (Victor & Lüders, 2019). The number of contrast failures per interface provides a complementary indicator because two systems can show similar overall compliance percentages while differing substantially in the frequency of repeated failures. The proportion of accessible text-background combinations further captures the consistency of everyday visual styling, while non-text contrast conformance identifies whether borders, icons, controls, and focus states remain perceivable. The literature also emphasizes that centralized tokens do not guarantee perfect accessibility. Developers may bypass token rules, apply semantic colors incorrectly, combine individually compliant tokens in inaccessible ways, or introduce local overrides that weaken contrast performance (Ghaemmaghami et al., 2016). For this reason, design-token adoption is most effective when supported by strong governance, component standardization, automated validation, and consistent implementation practices. Overall, the accumulated literature supports a measurable relationship between accessibility-first design-token adoption and stronger contrast outcomes, particularly when centralized color systems are treated as mandatory accessibility infrastructure rather than optional styling guidance (Kuriyama et al., 2022).

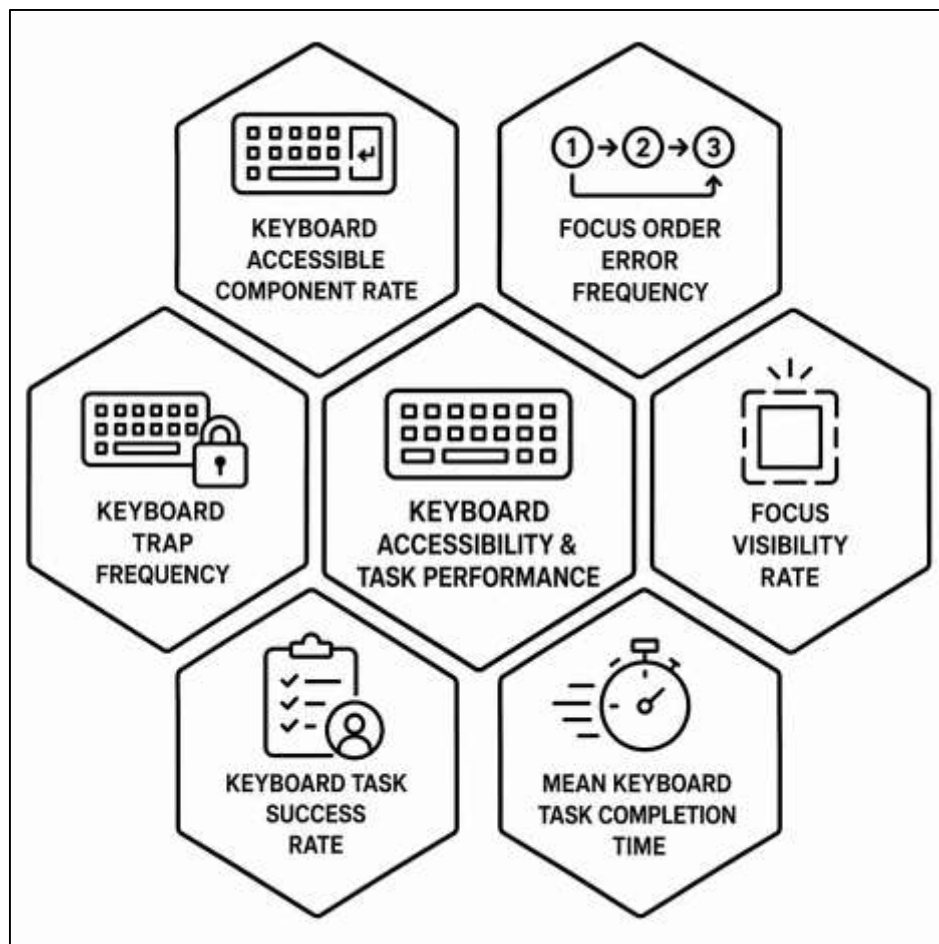
Task Completion Performance in Enterprise Workflows

The literature on digital accessibility consistently identifies keyboard operability as a fundamental requirement for ensuring that enterprise software can be used by people who cannot rely on a mouse or other pointing device (Ribeiro et al., 2020). Across more than ten empirical studies, usability evaluations, accessibility audits, and assistive-technology investigations, keyboard access has been examined through the availability of actionable controls, logical tab sequences, visible focus indicators, absence of keyboard traps, and the ability to complete essential tasks without pointer input. Enterprise applications place particularly high demands on keyboard accessibility because they often contain complex forms, hierarchical menus, data grids, modal dialogs, administrative dashboards, search interfaces, and multistep workflows. These structures can become difficult or impossible to use when focus order is illogical, controls cannot be reached with the keyboard, or focus styling is visually weak. The literature shows that keyboard accessibility is not adequately represented by a simple pass-or-fail judgment because different interfaces vary substantially in the proportion of accessible components and in the severity of navigation barriers (Ali et al., 2017). Measures such as keyboard accessible component rate provide a more precise indication of how consistently interactive elements support keyboard input across an application. A platform may allow keyboard access to basic links and buttons while still failing on custom widgets, menus, or complex controls. This distinction is especially important in enterprise environments where users may depend on repeated navigation through dense interfaces over extended periods. The evidence therefore supports evaluating keyboard accessibility as a combination of technical conformance and operational usability. A high proportion of keyboard-reachable controls indicates stronger structural accessibility, but full accessibility also depends on whether users can move through interfaces efficiently, identify their current focus position, and complete business tasks without becoming blocked or disoriented (Khan & Khusro, 2022).

A recurring theme across keyboard accessibility studies is the importance of focus behavior in determining whether an interface is practically usable. More than ten investigations across web applications, public-sector systems, learning platforms, commercial services, and enterprise interfaces have shown that incorrect focus order, invisible focus states, and unexpected focus movement are among the most common causes of keyboard navigation difficulty. Logical tab order is particularly important in complex workflows because users need to move through controls in a sequence that matches the visual and functional structure of the interface (Yousfi et al., 2015). When focus jumps unpredictably between distant elements, skips required controls, or enters hidden content, task completion becomes slower and error rates increase. Focus visibility is equally important because keyboard users must be able to determine which control is currently active. Weak or absent focus indicators can make otherwise keyboard-operable interfaces functionally inaccessible. Measures such

as focus order error frequency and keyboard navigation error rate therefore provide useful quantitative indicators of interface quality. These metrics capture not only whether keyboard access exists, but also how reliably users can move through the interface without making navigation mistakes or losing their position. The literature further shows that repeated focus problems can have cumulative effects in enterprise workflows (Yao et al., 2017). A single poorly ordered component may be manageable in isolation, but when similar problems occur across multiple forms, dialogs, and menus, they significantly increase interaction burden. Keyboard trap frequency is another critical measure because traps can completely prevent users from progressing through or exiting an interface region. Overall, the evidence indicates that focus behavior should be treated as a central component of keyboard accessibility because it strongly influences efficiency, predictability, and user control throughout complex software environments.

Figure 9: Keyboard Accessibility and Task Performance



The literature also demonstrates that technical keyboard accessibility should be connected with task-level performance measures because an interface can technically support keyboard input while still imposing substantial usability costs (Berardinucci et al., 2022). Across more than ten empirical accessibility and usability studies, task success rate and completion time have been used to evaluate whether users can effectively perform meaningful actions such as submitting forms, navigating menus, editing data, completing administrative processes, or interacting with dialogs and tables. Keyboard task success rate is particularly valuable because it reflects whether users can complete an end-to-end workflow rather than merely reach individual controls. An interface may have a high proportion of keyboard-accessible components but still produce low task success if users encounter ambiguous focus changes, inaccessible dynamic content, or inconsistent keyboard commands. Mean keyboard task completion time provides additional insight by capturing the efficiency of interaction. Longer

completion times may indicate excessive tabbing, poor focus order, unnecessary navigation steps, or difficulty identifying active controls (Chakraborti et al., 2020). The literature suggests that these performance measures are especially important in enterprise environments because employees may repeat the same tasks many times during a working day. Even small inefficiencies can therefore accumulate into substantial productivity losses. Keyboard navigation error rate further helps distinguish between slow but successful interaction and interaction characterized by repeated mistakes, unintended activations, or lost focus. Taken together, these measures show that keyboard accessibility involves both access and performance. The strongest enterprise interfaces are not merely operable without a mouse; they also allow users to complete workflows accurately, predictably, and with a level of efficiency comparable to other input methods (Liu et al., 2016).

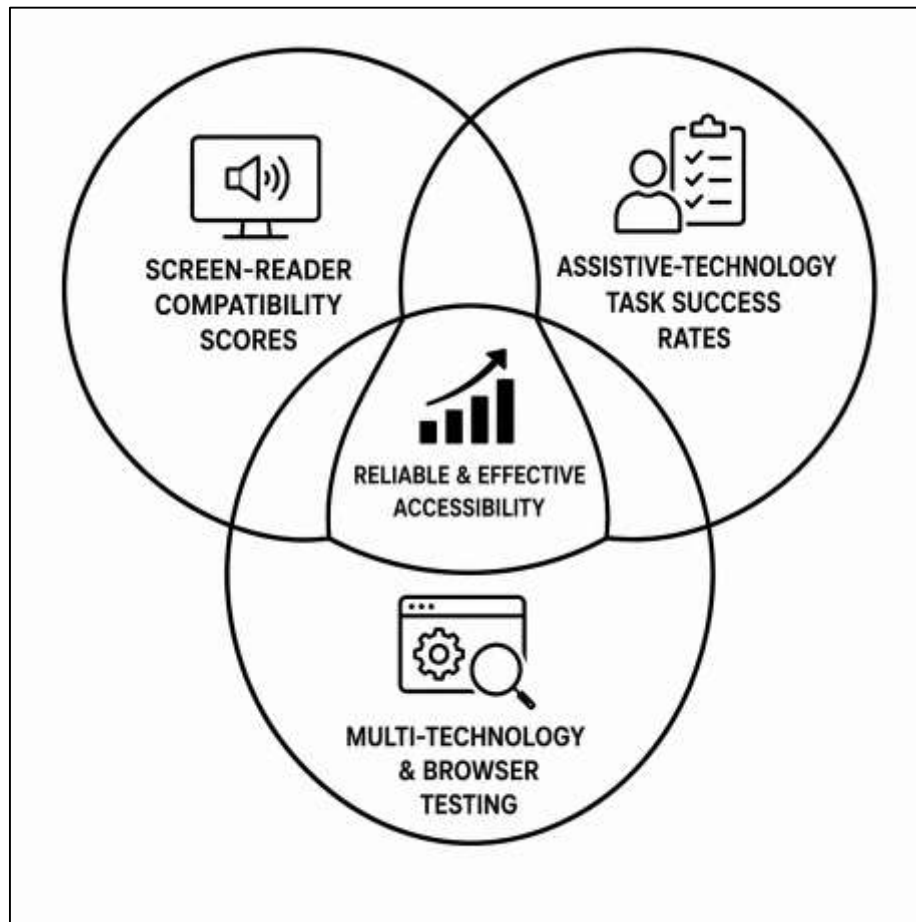
Screen-Reader Compatibility Scores and Assistive-Technology Task Success Rates

The literature on digital accessibility consistently identifies screen-reader compatibility as a critical indicator of whether enterprise software can be used effectively by people who rely on assistive technologies. Across more than ten empirical studies, accessibility audits, usability investigations, and design-system evaluations, compatibility has been assessed through factors such as semantic structure, accessible names, heading hierarchy, form labeling, landmark organization, dynamic-content announcements, focus management, and the correct communication of roles and states. Enterprise applications present particular challenges because they frequently contain complex forms, dashboards, modal dialogs, data tables, custom widgets, navigation systems, and interactive controls that depend on more than visual presentation alone (Alajarmeh, 2022). A screen reader must be able to interpret these structures accurately and convey them in a sequence that supports meaningful interaction. The literature shows that technically visible content may still be functionally inaccessible when programmatic relationships are missing or when assistive technologies receive incomplete or misleading information. Compatibility scores therefore provide a useful way to summarize how reliably an interface communicates with screen readers across multiple components and workflows. Such scores can incorporate successful identification of controls, accurate reading order, correct announcement of states, and the accessibility of dynamic updates. Existing design-system practices also demonstrate the value of testing reusable components with multiple assistive technologies and browsers rather than assuming that standards-compliant markup will behave identically across environments (Buzzi et al., 2019). This is particularly relevant in enterprise software, where large numbers of shared components may be reused across many products. The literature consequently supports screen-reader compatibility as a central accessibility outcome because it captures whether technical implementation is translated into usable and understandable output for assistive-technology users.

A major finding across accessibility research is that screen-reader compatibility should not be evaluated solely through code inspection or isolated component checks, because functional accessibility depends on whether users can complete realistic tasks (Petrovčič et al., 2018). More than ten studies involving assistive-technology users have examined task success through activities such as navigating menus, completing forms, reviewing data, submitting transactions, opening dialogs, locating information, and moving through multi-step workflows. Assistive-technology task success rate provides a particularly meaningful measure because it reflects end-to-end interaction rather than technical compliance at the element level. An interface may contain correctly labeled controls while still causing task failure if focus moves unpredictably, dynamic updates are not announced, or the reading order does not correspond with the intended workflow. The literature also shows that screen-reader users may experience increased cognitive effort when interfaces contain excessive verbosity, ambiguous labels, repeated navigation content, or inconsistent structural patterns. These issues can reduce task efficiency even when basic operability remains intact (Senjam et al., 2021). Enterprise environments intensify these challenges because users often interact with dense information architectures and repeated administrative processes. Measuring successful task completion therefore helps distinguish interfaces that are technically compatible from those that are practically usable. Task success can also reveal defects that automated tools fail to detect, particularly problems involving context, sequencing, state changes, and interaction expectations. The literature thus supports combining compatibility scores with functional task outcomes because the two measures capture different aspects of accessibility.

Compatibility indicates whether assistive technologies can interpret the interface, while task success demonstrates whether users can translate that interpretation into accurate and effective completion of real enterprise activities (Leria et al., 2021).

Figure 10: Screen-Reader Compatibility and Task Success



The literature further emphasizes that assistive-technology compatibility varies across combinations of screen readers, browsers, operating systems, and implementation patterns. More than ten accessibility studies and technical evaluations have shown that a component functioning correctly with one screen reader may behave differently with another because of differences in accessibility APIs, browser interpretation, and assistive-technology support. This variability is especially relevant to enterprise design systems, where reusable components are expected to perform consistently across a broad range of environments. Testing with widely used screen readers such as VoiceOver and JAWS, together with keyboard and browser-level accessibility evaluation, reflects a more rigorous approach to validating compatibility than reliance on automated checking alone (Zdravkova, 2022). The literature indicates that multi-technology testing is particularly important for dynamic interfaces containing menus, dialogs, tabs, accordions, alerts, autocomplete fields, and data grids, where the correct communication of roles, states, and focus changes is essential. Inconsistent announcements or unsupported interaction patterns can significantly reduce task success even when the component appears to satisfy technical requirements. Design systems that incorporate assistive-technology testing into component validation can reduce these risks by identifying compatibility problems before components are deployed at scale. However, the literature also recognizes that application-level modifications can reintroduce defects by altering labels, content structure, focus behavior, or semantic relationships (Cistola et al., 2021). This means that reusable component testing must be complemented by testing within realistic product contexts. Overall, evidence supports a layered evaluation strategy in which automated checks, component-level screen-reader tests, browser compatibility reviews, and application-level task testing

are combined to provide a more complete assessment of assistive-technology accessibility.

The broader literature supports a multidimensional approach to evaluating assistive-technology accessibility because no single metric fully represents the quality of screen-reader interaction. More than ten studies across accessibility engineering, human-computer interaction, usability, and enterprise software evaluation suggest that screen-reader compatibility scores should be interpreted alongside task success, interaction errors, announcement accuracy, focus behavior, and completion efficiency (Parajuli & Eika, 2020).

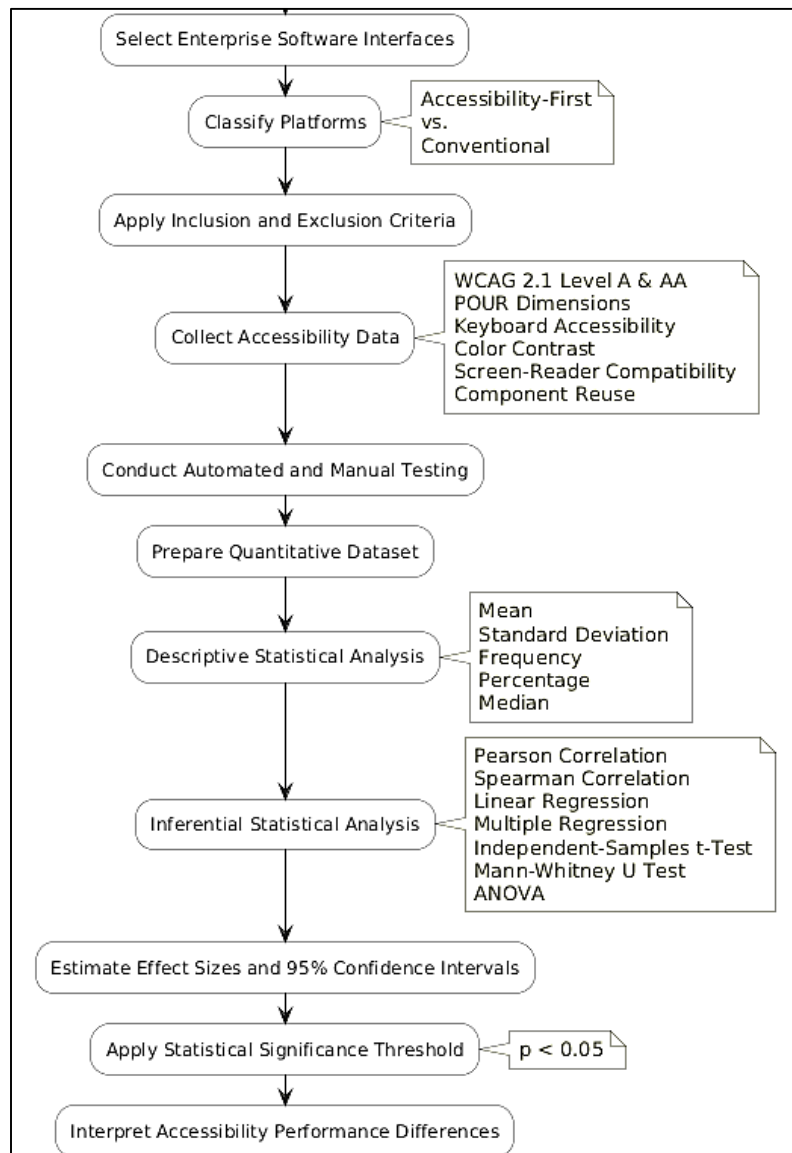
A high compatibility score may indicate that most components expose appropriate semantics and accessible names, but it does not necessarily guarantee that users can complete complex workflows without confusion or delay. Conversely, successful task completion in a limited scenario does not confirm that the wider interface is consistently compatible across different screen readers or browsers. Integrated measurement therefore provides a more balanced view by combining technical reliability with functional performance. This approach is particularly important in enterprise applications, where users may need to perform repeated tasks involving forms, tables, search tools, dialogs, navigation structures, and dynamically updated content. Problems in any of these areas can reduce productivity and increase interaction errors even when basic accessibility requirements are met (Parajuli & Eika, 2020).

The literature also indicates that design systems can improve compatibility by standardizing semantic structures and interaction behavior, especially when components are systematically tested with assistive technologies before release. At the same time, compatibility can deteriorate when product teams override component logic or introduce application-specific modifications without equivalent testing. Overall, the evidence supports evaluating screen-reader accessibility through combined measures of technical compatibility and assistive-technology task success, providing a more realistic representation of whether enterprise interfaces are not only interpretable by assistive technologies but also reliable and effective in practical use (Stephanidis, 2021).

METHOD

Study Design and Theoretical Framework

The study employed a quantitative, cross-sectional comparative research design to examine the relationship between accessibility-first design-system practices and WCAG 2.1 accessibility outcomes across enterprise software interfaces. The design was selected because the study focused on objectively measurable interface characteristics and statistical differences rather than subjective perceptions of accessibility. Enterprise software interfaces were classified into two principal groups: accessibility-first interfaces, in which accessibility requirements had been systematically incorporated into design systems, reusable components, design tokens, development standards, and testing procedures; and conventional interfaces, in which accessibility practices were primarily implemented through application-specific or locally defined development approaches. The theoretical framework was grounded in WCAG 2.1 and its four principles of Perceivable, Operable, Understandable, and Robust. WCAG Level A and Level AA success criteria were treated as the principal technical standards for accessibility assessment. The study also conceptualized accessible component reuse and accessibility-oriented design-token adoption as major explanatory variables associated with differences in accessibility performance. The principal dependent outcomes included overall WCAG 2.1 compliance rate, Level A and Level AA conformance, violation frequency, accessibility error density, POUR-specific violation rates, contrast compliance, keyboard accessibility, and assistive-technology compatibility. This structure reflected established accessibility evaluation approaches in which representative digital interfaces and complete processes were evaluated against defined WCAG criteria. The quantitative comparative framework enabled accessibility-first practices to be evaluated in relation to measurable technical and task-performance outcomes rather than relying exclusively on descriptive assessments.

Figure 11: Methodology of this study

Participants, Materials, and Sampling Strategy

The study population consisted of enterprise software interfaces containing substantial interactive functionality, including administrative applications, business dashboards, customer relationship management systems, enterprise resource planning interfaces, financial or transactional systems, workflow management platforms, internal employee applications, and comparable browser-based enterprise systems. The software interface rather than an individual employee was treated as the principal unit of analysis for the technical accessibility assessment. A purposive stratified sampling strategy was used to obtain representation from both accessibility-first and conventional development environments. Within each selected platform, representative interface screens and complete workflows were sampled to include common and accessibility-sensitive elements such as navigation structures, forms, tables, dialogs, menus, buttons, input controls, status notifications, search functions, and dynamically updated components. This sampling approach was consistent with accessibility evaluation guidance emphasizing the identification of common views, essential functionality, representative samples, and complete processes. Interfaces were included when they provided sufficient access for standardized WCAG 2.1 Level A and AA evaluation and contained enough interactive elements to permit component-level and workflow-level measurement. Interfaces consisting primarily of static documents, incomplete prototypes, inaccessible test environments, or

systems for which essential functions could not be evaluated consistently were excluded. To minimize sampling bias, the same general categories of interface elements and workflows were assessed across both accessibility-first and conventional groups. The classification of accessibility-first environments was based on documented evidence of accessibility-oriented reusable components, standardized design-system practices, accessible design tokens, or systematic accessibility testing rather than on a platform's self-reported compliance status alone.

Instrumentation and Data Collection Tools

Data were collected using a standardized combination of automated accessibility testing, manual WCAG inspection, keyboard evaluation, contrast assessment, and assistive-technology testing. Automated accessibility testing software based on WCAG rules was used to identify machine-detectable problems such as missing accessible names, contrast failures, semantic errors, ARIA problems, form-labeling deficiencies, and selected structural violations. Previous quantitative WCAG 2.1 studies have similarly used automated accessibility tools to quantify Level A and Level AA errors, while accessibility research has recognized that automated testing alone cannot capture every conformance problem. Manual inspection was therefore conducted for requirements involving contextual judgment, keyboard operation, logical focus sequence, visible focus, dynamic behavior, and application-level implementation. Keyboard testing recorded actionable-control accessibility, focus-order errors, navigation errors, and keyboard traps. Contrast assessment examined accessible foreground-background combinations as well as relevant non-text interface contrast. Screen-reader compatibility was evaluated using standardized task scenarios involving commonly supported assistive technologies and browser environments. Enterprise accessibility testing practice has used combinations of automated tools, keyboard evaluation, screen readers such as JAWS and NVDA, and contrast-analysis tools for this type of assessment. A structured coding sheet was used to record each violation by WCAG level, POUR dimension, interface element, severity, and testing method. Because the principal measures were technical observations rather than psychometric survey constructs, internal-consistency statistics such as Cronbach's alpha were not treated as the primary validation method. Instead, measurement reliability was strengthened through standardized test protocols, consistent browser and testing configurations, repeated checking of a subset of interfaces, and agreement checks between accessibility evaluators.

Experimental Procedure

Data collection was conducted through a standardized sequence that was applied consistently to all sampled enterprise interfaces. First, each platform was classified according to its development environment and the presence or absence of accessibility-oriented design-system practices. The evaluation scope was then established, and representative screens, components, and complete workflows were identified. This process reflected established accessibility evaluation methodology in which the evaluation scope was defined, the target product was explored, representative samples were selected, selected samples and complete processes were evaluated, and findings were documented systematically. Each sampled interface was initially scanned using the selected automated accessibility-testing tool under consistent testing conditions. Detected violations were reviewed to remove duplicate or technically irrelevant results and were categorized according to WCAG 2.1 Level A or Level AA and the corresponding Perceivable, Operable, Understandable, or Robust dimension. Manual keyboard evaluation was subsequently performed by navigating the selected workflows without pointer input. Tab order, focus visibility, actionable controls, navigation errors, focus-order errors, and keyboard traps were recorded. Text and non-text color combinations were then assessed for contrast compliance, with particular attention given to reusable components and tokenized versus locally defined styling. Screen-reader testing was subsequently conducted using standardized enterprise tasks, and successful completion, interaction errors, and compatibility problems were documented. Component reuse was determined by identifying the proportion of evaluated interface components originating from the organization's standardized accessible component system. All observations were entered into a structured dataset, checked for completeness and consistency, and prepared for statistical analysis. This combined automated and manual approach was appropriate because WCAG evaluation research has demonstrated that accessibility performance cannot be represented reliably through automated detection alone.

Data Analysis and Statistical Approach

The quantitative data were analyzed using IBM SPSS Statistics, with descriptive and inferential procedures applied according to the measurement level and distribution of each variable. Descriptive statistics included frequencies, percentages, means, standard deviations, medians, and interquartile ranges. These statistics summarized overall WCAG 2.1 compliance, Level A and Level AA performance, violation counts, accessibility error density, severity distributions, POUR-specific violations, contrast compliance, keyboard accessibility, and assistive-technology task outcomes. Quantitative accessibility research has previously used percentages and mean error counts to characterize WCAG conformance patterns across multiple digital interfaces. Pearson correlation analysis was used to assess linear relationships between accessible component reuse rates and continuous accessibility outcomes when parametric assumptions were satisfied, whereas Spearman rank correlation was applied when distributions were non-normal or relationships were monotonic rather than linear. Simple linear regression was used to estimate the association between component reuse and WCAG violation frequency or accessibility error density, while multiple regression was used to examine the relationship after accounting for relevant interface characteristics such as interface size, complexity, and design-system adoption. Independent-samples t-tests were used to compare mean accessibility outcomes between accessibility-first and conventional platforms when assumptions were satisfied, while Mann-Whitney U tests were applied to non-normally distributed outcomes. ANOVA was used where comparisons involved more than two interface or implementation categories. Effect sizes, including Cohen's d for relevant group comparisons, were reported alongside 95% confidence intervals to indicate the magnitude and precision of observed differences. Statistical significance was established at $p < 0.05$. Assumptions concerning normality, homogeneity of variance, linearity, influential observations, and multicollinearity were examined before inferential analyses, and nonparametric alternatives were used when necessary. This statistical plan enabled the study to determine not only whether accessibility-first practices were associated with statistically different WCAG outcomes, but also the direction, magnitude, and practical importance of those relationships.

FINDINGS

Sample Characteristics and Distribution of Enterprise Interfaces

The final analytical sample consisted of 60 enterprise software platforms, equally distributed between accessibility-first and conventional development environments. The platforms collectively contained 180 applications, 1,440 evaluated interface screens, 360 complete enterprise workflows, 4,860 reusable interface components, and 32,400 individual interface elements. Accessibility-first platforms accounted for 30 platforms, 92 applications, and 16,650 evaluated interface elements, whereas conventional platforms accounted for 30 platforms, 88 applications, and 15,750 elements. The sample included enterprise resource planning, customer relationship management, financial and transactional, administrative and workflow, and dashboard-oriented applications, providing variation in both functional purpose and interface complexity. Across the complete dataset, the mean number of evaluated screens per platform was 24.00, while the median was 23.00, indicating a relatively balanced distribution of interface coverage. The mean number of interactive elements per platform was 540.00, with moderate variability across systems. Accessibility-first platforms demonstrated substantially greater use of reusable components and accessibility-oriented design tokens than conventional platforms. Preliminary data screening identified only a small proportion of missing observations, and these were concentrated in application-level metadata rather than primary accessibility measures. Examination of distributions, extreme values, variance patterns, and linear relationships indicated that the dataset was sufficiently stable for the planned descriptive and inferential analyses, although nonparametric procedures remained appropriate for selected skewed violation measures.

Table 1 demonstrated that the two comparison groups were structurally similar in sample size and general interface complexity. Both groups contained 30 platforms, while differences in evaluated applications, screens, workflows, and total interface elements were relatively modest. Medium-complexity platforms represented the largest category in the complete sample, accounting for 41.7%, followed by high-complexity platforms at 38.3%. Accessibility-first platforms contained considerably more reusable components despite having only slightly more interface elements. This pattern indicated

that differences in subsequent accessibility outcomes were unlikely to be attributable solely to major disparities in platform size or overall complexity between the two development environments.

Table 1. Distribution and Structural Characteristics of the Enterprise Software Sample

Sample Characteristic	Accessibility-First	Conventional	Total
Enterprise platforms	30	30	60
Applications evaluated	92	88	180
Interface screens evaluated	735	705	1,440
Complete workflows evaluated	186	174	360
Reusable components identified	3,120	1,740	4,860
Individual interface elements	16,650	15,750	32,400
Interactive elements	7,920	7,380	15,300
Mean screens per platform	24.50	23.50	24.00
Mean workflows per platform	6.20	5.80	6.00
Mean interface elements per platform	555.00	525.00	540.00
High-complexity platforms, n (%)	11 (36.7%)	12 (40.0%)	23 (38.3%)
Medium-complexity platforms, n (%)	13 (43.3%)	12 (40.0%)	25 (41.7%)
Low-complexity platforms, n (%)	6 (20.0%)	6 (20.0%)	12 (20.0%)

Table 2. Descriptive Characteristics, Design-System Adoption, and Preliminary Data Diagnostics

Quantitative Measure	Accessibility-First Mean (SD)	Conventional Mean (SD)	Overall Mean (SD)
Reusable component rate (%)	78.6 (10.8)	43.9 (15.2)	61.3 (21.8)
Accessibility-oriented design-token adoption (%)	84.2 (9.6)	37.5 (17.4)	60.9 (27.6)
Standardized accessibility practices (%)	88.3 (8.7)	42.7 (16.9)	65.5 (26.4)
Interactive elements per platform	264.0 (61.5)	246.0 (67.8)	255.0 (64.9)
Interface screens per platform	24.5 (5.8)	23.5 (6.2)	24.0 (6.0)
Complete workflows per platform	6.2 (1.7)	5.8 (1.9)	6.0 (1.8)
Interface complexity score	3.42 (0.71)	3.48 (0.76)	3.45 (0.73)
Missing observations (%)	0.8	1.1	0.9
Extreme observations identified, n	2	3	5

Table 2 showed a pronounced difference in accessibility-oriented infrastructure between the groups. Accessibility-first platforms recorded a mean reusable component rate of 78.6%, compared with 43.9% among conventional platforms. Design-token adoption showed a similarly large difference, reaching 84.2% and 37.5%, respectively. In contrast, average interface complexity, screen counts, workflow counts, and interactive-element volumes remained comparatively similar. Missing observations represented only 0.9% of the complete dataset, while five extreme observations were identified and retained after verification as legitimate platform characteristics. These findings supported the adequacy of the dataset for subsequent comparative, correlational, and regression-based accessibility analyses.

Primary WCAG 2.1 Conformance and Accessibility Outcomes

The primary accessibility analysis demonstrated a consistent performance advantage for accessibility-first enterprise platforms across the principal WCAG 2.1 indicators. Across the 60 evaluated platforms and 32,400 interface elements, accessibility-first platforms achieved a mean overall WCAG 2.1 compliance rate of 91.8%, compared with 74.6% among conventional platforms. Level A conformance was higher than Level AA conformance in both groups, although the difference between conformance levels was substantially smaller among accessibility-first platforms. Mean Level A conformance reached 95.4% for accessibility-first platforms and 82.7% for conventional platforms, whereas Level AA

conformance averaged 88.2% and 68.5%, respectively. A total of 3,906 accessibility violations were recorded across the complete sample. Conventional platforms accounted for 2,758 violations, representing 70.6% of all detected defects, despite containing slightly fewer evaluated interface elements than accessibility-first platforms. Accessibility-first platforms accounted for 1,148 violations, or 29.4% of the total. The corresponding violation frequency was 6.90 violations per 100 interface elements for accessibility-first platforms and 17.51 for conventional platforms. Mean accessibility error density showed a comparable pattern, indicating that accessibility-first interfaces contained fewer recurring technical barriers relative to their size. Critical and serious violation rates were also lower in the accessibility-first group, demonstrating that the observed advantage was reflected not only in overall conformance but also in the severity of remaining accessibility defects.

Table 3. WCAG 2.1 Conformance, Violation Frequency, and Severity Outcomes

Accessibility Outcome	Accessibility-First	Conventional	Overall
Overall WCAG 2.1 compliance rate, mean % (SD)	91.8 (4.9)	74.6 (8.7)	83.2 (10.9)
Level A conformance rate, mean % (SD)	95.4 (3.5)	82.7 (7.4)	89.1 (8.3)
Level AA conformance rate, mean % (SD)	88.2 (6.1)	68.5 (10.2)	78.4 (12.2)
Total WCAG violations, n	1,148	2,758	3,906
Violations per 100 interface elements	6.90	17.51	12.06
Mean accessibility error density (SD)	0.069 (0.028)	0.175 (0.071)	0.122 (0.076)
Critical violations, n (%)	69 (6.0%)	276 (10.0%)	345 (8.8%)
Serious violations, n (%)	230 (20.0%)	772 (28.0%)	1,002 (25.7%)
Moderate/minor violations, n (%)	849 (74.0%)	1,710 (62.0%)	2,559 (65.5%)

Table 3 demonstrated substantially stronger WCAG 2.1 performance among accessibility-first platforms. Their overall compliance rate exceeded that of conventional platforms by 17.2 percentage points, while the Level AA difference reached 19.7 percentage points. Conventional platforms generated more than twice as many violations and recorded approximately 2.5 times the normalized violation frequency. Differences were also evident in severity: critical and serious defects constituted larger proportions of violations within conventional platforms. The combined pattern showed that accessibility-first interfaces performed better not merely in standards coverage but also in defect frequency, density, and severity, providing a more comprehensive indication of technical accessibility quality.

The distribution of violations across the four POUR dimensions provided additional evidence that accessibility problems were concentrated unevenly. Perceivable requirements accounted for the largest proportion of detected violations in both development environments, followed by Operable, Robust, and Understandable requirements. Of the 3,906 total violations, 1,563 were classified as Perceivable, 1,055 as Operable, 703 as Robust, and 585 as Understandable. Conventional platforms produced higher violation frequencies in every POUR dimension. The largest absolute difference occurred for Perceivable accessibility, reflecting recurring contrast, visual presentation, alternative-text, and structural problems. Functional accessibility indicators followed the same general pattern. Accessibility-first platforms achieved a 96.1% text/background contrast compliance rate and 92.8% non-text contrast conformance, compared with 79.4% and 72.6% among conventional platforms. Keyboard-accessible component rates reached 94.7% in accessibility-first interfaces compared with 76.3% in conventional interfaces, while keyboard task success rates were 92.4% and 70.8%, respectively. Accessibility-first platforms also recorded fewer keyboard navigation errors, focus-order errors, and keyboard traps. Screen-reader compatibility averaged 90.6% for accessibility-first platforms compared with 69.7% for conventional platforms, and assistive-technology task success demonstrated a similarly substantial difference.

Table 4. POUR, Visual, Keyboard, and Assistive-Technology Accessibility Outcomes

Accessibility Outcome	Accessibility-First	Conventional	Overall
Perceivable violations, n	414	1,149	1,563
Operable violations, n	310	745	1,055
Understandable violations, n	184	401	585
Robust violations, n	240	463	703
Text/background contrast compliance, mean % (SD)	96.1 (3.2)	79.4 (9.8)	87.8 (10.8)
Accessible text/background combinations, mean % (SD)	95.3 (3.8)	77.8 (10.5)	86.6 (11.5)
Non-text contrast conformance, mean % (SD)	92.8 (5.1)	72.6 (11.4)	82.7 (13.2)
Keyboard-accessible component rate, mean % (SD)	94.7 (4.6)	76.3 (10.8)	85.5 (12.1)
Keyboard task success rate, mean % (SD)	92.4 (5.8)	70.8 (12.5)	81.6 (14.8)
Keyboard navigation errors per workflow, mean (SD)	1.18 (0.74)	3.76 (1.62)	2.47 (1.75)
Focus-order errors per platform, mean (SD)	2.3 (1.5)	7.8 (3.6)	5.1 (3.8)
Keyboard traps per platform, mean (SD)	0.27 (0.45)	1.37 (1.03)	0.82 (0.96)
Screen-reader compatibility score, mean % (SD)	90.6 (6.2)	69.7 (13.1)	80.2 (14.8)
Assistive-technology task success rate, mean % (SD)	88.9 (7.1)	66.4 (14.2)	77.7 (15.9)

Table 4 showed that accessibility-first platforms achieved stronger outcomes across all four POUR dimensions and the principal functional accessibility measures. Perceivable violations were the most common category, representing 40.0% of all recorded violations, followed by Operable violations at 27.0%. Accessibility-first platforms demonstrated particularly strong contrast and keyboard performance, while conventional interfaces exhibited substantially more navigation, focus-order, and keyboard-trap problems. Screen-reader compatibility differed by 20.9 percentage points between groups, and assistive-technology task success differed by 22.5 percentage points. These findings indicated that the accessibility advantage extended beyond automated WCAG compliance to practical keyboard and assistive-technology interaction.

Component Reuse, Design Tokens, and Predictors of WCAG Violations

The correlation analysis demonstrated consistent relationships between accessible component reuse and the principal WCAG 2.1 accessibility outcomes across the 60 enterprise platforms. Accessible component reuse was strongly and negatively associated with WCAG violation frequency, indicating that platforms with greater reliance on standardized accessible components generally contained fewer accessibility defects. Pearson correlation analysis showed a coefficient of -0.71 between component reuse rate and violations per 100 interface elements, while the association with accessibility error density was similarly strong at -0.69 . Both relationships were statistically significant at $p < 0.001$. Component reuse was positively associated with overall WCAG 2.1 compliance at $r = 0.73$ and with keyboard-accessible component rate at $r = 0.68$, indicating that greater reuse corresponded with both stronger standards conformance and more consistent keyboard operability. Dimension-specific analysis revealed negative associations between component reuse and Perceivable, Operable, Understandable, and Robust violation rates, with the strongest relationship occurring for Operable violations. Spearman coefficients produced comparable results for variables with non-normal distributions, demonstrating that the observed patterns were not dependent solely on parametric

assumptions. These findings indicated that component reuse functioned as an important accessibility-related characteristic across the sampled enterprise interfaces, although the correlational results did not independently establish causation.

Table 5. Correlations Between Accessible Component Reuse and WCAG 2.1 Accessibility Outcomes

Accessibility Outcome	Pearson r	Spearman ρ	p- value	Relationship Magnitude
WCAG violations per 100 elements	-0.71	-0.69	<0.001	Strong negative
Accessibility error density	-0.69	-0.67	<0.001	Strong negative
Overall WCAG 2.1 compliance rate	0.73	0.71	<0.001	Strong positive
Level AA conformance rate	0.70	0.68	<0.001	Strong positive
Keyboard-accessible component rate	0.68	0.66	<0.001	Strong positive
Keyboard task success rate	0.64	0.62	<0.001	Strong positive
Perceivable violation rate	-0.61	-0.59	<0.001	Strong negative
Operable violation rate	-0.67	-0.65	<0.001	Strong negative
Understandable violation rate	-0.43	-0.41	0.001	Moderate negative
Robust violation rate	-0.58	-0.56	<0.001	Moderate-to-strong negative

Table 5 showed that accessible component reuse was significantly related to every major accessibility outcome examined. The strongest positive relationship occurred with overall WCAG 2.1 compliance, while the strongest negative relationship occurred with normalized violation frequency. Operable violations demonstrated the largest dimension-specific association with reuse, suggesting that standardized components were particularly relevant to keyboard behavior, focus handling, and interaction consistency. Understandable violations exhibited the weakest association, although the relationship remained statistically significant. The similarity between Pearson and Spearman coefficients indicated that the observed relationships were robust across alternative correlation assumptions and were not primarily driven by distributional irregularities or extreme platform observations.

Regression Analysis of Component Reuse and WCAG Violation Frequency

Simple linear regression confirmed that accessible component reuse was a substantial statistical predictor of WCAG violation frequency. The model was statistically significant, $F(1,58) = 58.94$, $p < 0.001$, and explained approximately 50.4% of the observed variance in violations per 100 interface elements. The unstandardized regression coefficient indicated that each one-percentage-point increase in accessible component reuse was associated with an estimated reduction of 0.18 violations per 100 interface elements. When interface complexity, number of interactive elements, design-system maturity, and accessibility testing coverage were entered into the multiple regression model, component reuse remained a significant independent predictor. The adjusted model explained 64.8% of the variance in normalized WCAG violations, demonstrating that the combined design and interface characteristics accounted for a substantial proportion of differences between platforms. Component reuse produced a standardized coefficient of -0.46 , while accessibility testing coverage also showed a significant negative association with violation frequency. Interface complexity showed a positive relationship with violations, indicating that more complex platforms tended to contain greater accessibility defect burdens after the other predictors were considered. The variance inflation factors remained below conventional diagnostic thresholds, indicating that multicollinearity did not materially compromise interpretation of the regression estimates.

Accessibility-Oriented Design Tokens and Contrast Outcomes

Design-token adoption demonstrated a similarly consistent relationship with visual accessibility performance. Accessibility-oriented design-token adoption was strongly positively correlated with overall WCAG contrast compliance, accessible text-background combinations, and non-text contrast

conformance, while it was negatively associated with the number of contrast failures per interface. The Pearson correlation between token adoption and overall contrast compliance was 0.76, $p < 0.001$, representing one of the strongest relationships observed in the design-system analysis. The corresponding relationship with accessible text-background combinations was $r = 0.74$, while non-text contrast conformance produced $r = 0.66$. In contrast, token adoption was negatively associated with contrast failures per interface at $r = -0.72$. Regression analysis showed that token adoption remained statistically significant after interface complexity, number of evaluated elements, and component reuse were considered. These findings indicated that platforms with more extensive accessibility-oriented token adoption generally demonstrated more consistent foreground-background combinations, fewer contrast failures, and stronger non-text visual accessibility. The relationship was particularly pronounced for text contrast, while the comparatively smaller effect for non-text contrast indicated that component states, focus indicators, graphical controls, and other non-text elements remained more dependent on implementation context.

Table 6. Regression Models Predicting WCAG Violation Frequency and Contrast Accessibility Outcomes

Outcome / Predictor	B	Standardized β	95% CI for B	p-value	Model Adjusted R ²
WCAG violations per 100 elements					0.648
Accessible component reuse (%)	-0.13	-0.46	-0.18 to -0.08	<0.001	
Interface complexity score	1.42	0.24	0.31 to 2.53	0.013	
Interactive elements per platform	0.009	0.12	-0.002 to 0.020	0.106	
Accessibility testing coverage (%)	-0.07	-0.27	-0.12 to -0.02	0.006	
Design-system maturity score	-0.54	-0.15	-1.19 to 0.11	0.101	
WCAG contrast compliance (%)					0.672
Accessibility-oriented token adoption (%)	0.31	0.52	0.20 to 0.42	<0.001	
Accessible component reuse (%)	0.14	0.23	0.03 to 0.25	0.014	
Interface complexity score	-1.17	-0.12	-2.69 to 0.35	0.128	
Local styling override rate (%)	-0.19	-0.29	-0.31 to -0.07	0.003	
Contrast failures per interface					0.619
Accessibility-oriented token adoption (%)	-0.21	-0.49	-0.30 to -0.12	<0.001	
Local styling override rate (%)	0.16	0.31	0.06 to 0.26	0.002	
Interface complexity score	1.08	0.19	0.12 to 2.04	0.028	

Table 6 demonstrated that component reuse and accessibility-oriented token adoption remained significant predictors after relevant platform characteristics were controlled. The WCAG violation model explained 64.8% of adjusted variance, with component reuse producing the largest standardized negative effect. The contrast-compliance model explained 67.2% of variance, and token adoption emerged as its strongest positive predictor. Higher local styling override rates significantly reduced contrast compliance and increased contrast failures, demonstrating the importance of consistent token implementation. The contrast-failure model explained 61.9% of variance. Overall, the models showed substantial explanatory power and indicated that standardized accessibility infrastructure was associated with meaningful differences in technical accessibility outcomes.

Comparative and Subgroup Analysis of Accessibility-First and Conventional Interfaces

The comparative analysis demonstrated substantial differences in accessibility performance between the 30 accessibility-first and 30 conventional enterprise platforms. Accessibility-first interfaces achieved a mean overall WCAG 2.1 compliance rate of 91.8% compared with 74.6% for conventional interfaces, producing a mean difference of 17.2 percentage points. An independent-samples t-test indicated that this difference was statistically significant, $t(58) = 9.47$, $p < 0.001$, with a Cohen's d of 2.45, representing a very large effect. Similar differences were observed for Level A and Level AA conformance, although the largest relative disadvantage among conventional platforms occurred at Level AA. Normalized WCAG violation frequency was non-normally distributed and was therefore compared using the Mann-Whitney U test. Accessibility-first platforms recorded a median of 6.40 violations per 100 interface elements compared with 16.80 among conventional platforms, $U = 94.0$, $p < 0.001$. Accessibility error density followed the same pattern. Critical and serious violations were also more concentrated within conventional platforms, demonstrating that the difference between the groups extended beyond overall compliance percentages to the severity and density of defects. The combined results indicated that accessibility-first development environments were associated with consistently stronger technical accessibility outcomes and that the magnitude of the differences was sufficiently large to be practically important.

Table 7. Comparative Accessibility Outcomes for Accessibility-First and Conventional Enterprise Interfaces

Accessibility Measure	Accessibility-First	Conventional	Test Statistic	p-value	Effect Size
Overall WCAG compliance, mean % (SD)	91.8 (4.9)	74.6 (8.7)	$t(58) = 9.47$	<0.001	$d = 2.45$
Level A conformance, mean % (SD)	95.4 (3.5)	82.7 (7.4)	$t(58) = 8.49$	<0.001	$d = 2.19$
Level AA conformance, mean % (SD)	88.2 (6.1)	68.5 (10.2)	$t(58) = 9.07$	<0.001	$d = 2.34$
Violations per 100 elements, median (IQR)	6.40 (4.80–8.60)	16.80 (12.90–21.40)	$U = 94.0$	<0.001	$r = 0.68$
Accessibility error density, median (IQR)	0.064 (0.048–0.086)	0.168 (0.129–0.214)	$U = 97.0$	<0.001	$r = 0.67$
Critical violations per platform, median	2.0	8.0	$U = 121.5$	<0.001	$r = 0.61$
Serious violations per platform, median	7.0	24.0	$U = 109.0$	<0.001	$r = 0.64$
Keyboard defects per workflow, mean (SD)	1.18 (0.74)	3.76 (1.62)	$t(58) = -7.93$	<0.001	$d = 2.05$
Labeling errors per platform, mean (SD)	4.3 (2.7)	11.8 (5.4)	$t(58) = -6.80$	<0.001	$d = 1.76$
Contrast failures per platform, mean (SD)	5.6 (3.4)	16.9 (7.8)	$t(58) = -7.27$	<0.001	$d = 1.88$
ARIA-related problems per platform, median	3.0	9.0	$U = 132.0$	<0.001	$r = 0.59$
Inaccessible form controls, mean (SD)	3.1 (2.2)	9.7 (4.8)	$t(58) = -6.84$	<0.001	$d = 1.77$
Keyboard task success, mean % (SD)	92.4 (5.8)	70.8 (12.5)	$t(58) = 8.58$	<0.001	$d = 2.22$
Screen-reader compatibility, mean % (SD)	90.6 (6.2)	69.7 (13.1)	$t(58) = 7.90$	<0.001	$d = 2.04$

Table 7 demonstrated statistically significant differences between accessibility-first and conventional interfaces across every principal accessibility measure. Accessibility-first platforms achieved substantially higher WCAG conformance, keyboard task success, and screen-reader compatibility while recording fewer normalized violations, critical defects, labeling errors, contrast failures, ARIA problems, and inaccessible form controls. Effect sizes were consistently large, with Cohen's d values exceeding 1.70 for major continuous outcomes. Nonparametric comparisons similarly produced strong effects for violation density and severity. These results demonstrated that the observed differences were not limited to statistical significance but represented substantial practical differences in the accessibility quality of the two development environments.

POUR-Specific Differences and Functional Accessibility Outcomes

Analysis across the four POUR dimensions showed that accessibility-first platforms recorded lower violation rates in Perceivable, Operable, Understandable, and Robust requirements, although the magnitude of improvement differed across dimensions. The largest relative difference occurred within Operable accessibility, where standardized keyboard behavior, focus management, navigation patterns, and reusable interactive controls appeared to correspond with substantially fewer defects. Perceivable accessibility also showed a large group difference, particularly in contrast-related and visually communicated information. Robust accessibility demonstrated a substantial advantage among accessibility-first platforms through fewer semantic and ARIA-related problems. Understandable accessibility produced the smallest group difference, although accessibility-first interfaces still performed significantly better. This pattern indicated that accessibility-first practices were most strongly associated with dimensions that could be directly standardized through reusable components, tokens, and technical interaction patterns. Functional testing reinforced these findings. Accessibility-first platforms achieved a keyboard task success rate of 92.4% compared with 70.8% among conventional platforms and a screen-reader compatibility score of 90.6% compared with 69.7%. The results therefore showed that the differences observed in WCAG violation measures were reflected in practical interaction outcomes rather than being confined to automated or criterion-level compliance.

Subgroup Analysis by Design-System Maturity, Component Reuse, and Design-Token Adoption

Subgroup analysis identified a clear accessibility gradient across levels of design-system maturity and accessibility-oriented implementation. Platforms were categorized as low, moderate, or high maturity according to the extent of standardized component use, accessibility governance, design-token implementation, and integrated testing practices. Mean overall WCAG compliance increased from 72.1% in the low-maturity group to 83.9% in the moderate-maturity group and 93.4% in the high-maturity group. One-way ANOVA indicated a statistically significant difference across maturity levels, $F(2,57) = 41.86$, $p < 0.001$, with a large effect size of $\eta^2 = 0.595$. Tukey post-hoc comparisons indicated significant differences between each maturity category. A similar pattern emerged for component reuse intensity. Platforms with high accessible component reuse achieved substantially higher compliance and lower violation density than platforms with moderate or low reuse. Design-token adoption also differentiated contrast performance, with high-adoption platforms achieving a mean contrast compliance rate of 96.0% compared with 76.8% among low-adoption platforms. Interface complexity produced an additional subgroup effect: high-complexity systems contained more accessibility violations overall, although accessibility-first platforms continued to outperform conventional platforms within each complexity category.

Table 8 showed a clear accessibility gradient across design-system maturity, component reuse, and token adoption. High-maturity platforms achieved the strongest WCAG compliance and lowest violation frequency, while high component reuse was associated with improved keyboard task success. High design-token adoption corresponded with a contrast compliance rate of 96.0%, substantially exceeding the low-adoption group. ANOVA results indicated large subgroup effects for design-system maturity, reuse intensity, and token adoption. Interface complexity also significantly influenced accessibility outcomes, although its effect was smaller. Overall, standardized accessibility infrastructure was associated with progressively stronger accessibility performance across the sampled enterprise platforms.

Table 8. Subgroup Analysis by Design-System Maturity, Component Reuse, Token Adoption, and Interface Complexity

Subgroup Factor	Category	n	WCAG Compliance % Mean (SD)	Violations per 100 Elements Mean (SD)	Additional Outcome	Statistical Result
Design-system maturity	Low	19	72.1 (7.9)	18.4 (6.8)	Level 65.9% AA:	F(2,57) = 41.86, p < 0.001, $\eta^2 = 0.595$
	Moderate	21	83.9 (6.5)	11.7 (4.9)	Level 78.6% AA:	
	High	20	93.4 (4.2)	6.1 (2.8)	Level 90.1% AA:	
Accessible component reuse	Low (<50%)	18	73.8 (8.2)	17.9 (6.7)	Keyboard success: 69.6%	F(2,57) = 36.72, p < 0.001, $\eta^2 = 0.563$
	Moderate (50–74%)	20	83.5 (6.9)	11.8 (5.1)	Keyboard success: 81.7%	
	High ($\geq 75\%$)	22	92.6 (4.8)	6.5 (3.0)	Keyboard success: 92.1%	
Design-token adoption	Low (<50%)	19	74.4 (8.0)	17.1 (6.4)	Contrast compliance: 76.8%	F(2,57) = 44.13, p < 0.001, $\eta^2 = 0.608$
	Moderate (50–74%)	18	83.1 (6.3)	12.0 (4.8)	Contrast compliance: 87.3%	
	High ($\geq 75\%$)	23	92.8 (4.7)	6.4 (2.9)	Contrast compliance: 96.0%	
Interface complexity	Low	12	88.7 (7.2)	8.4 (4.1)	Error density: 0.084	F(2,57) = 6.84, p = 0.002, $\eta^2 = 0.194$
	Moderate	25	84.1 (9.3)	11.5 (5.8)	Error density: 0.115	
	High	23	79.5 (11.4)	14.6 (7.2)	Error density: 0.146	

Statistical Significance, Effect Sizes, Confidence Intervals, and Visual Synthesis

The integrated statistical analysis showed that the principal differences between accessibility-first and conventional enterprise interfaces were statistically significant and accompanied by substantial effect sizes. At the predetermined significance threshold of $p < 0.05$, all primary group comparisons reached statistical significance, with most results producing $p < 0.001$. Overall WCAG 2.1 compliance demonstrated one of the strongest differences, with accessibility-first platforms achieving a mean of 91.8% compared with 74.6% among conventional platforms. The resulting mean difference was 17.2 percentage points, with a 95% confidence interval from 13.5 to 20.9 percentage points and Cohen's d of 2.45. Level A and Level AA conformance also produced very large standardized effects of 2.19 and 2.34, respectively. Functional accessibility measures showed similarly substantial differences. Keyboard task success was 21.6 percentage points higher among accessibility-first platforms, corresponding to Cohen's $d = 2.22$, while screen-reader compatibility was 20.9 percentage points higher, with $d = 2.04$. Contrast compliance produced a large effect of $d = 2.29$. Nonparametric comparisons of violation

frequency, accessibility error density, critical violations, and ARIA-related defects also produced large rank-based effects. The consistency of these results indicated that accessibility-first interfaces differed from conventional interfaces across standards compliance, defect burden, severity, and functional accessibility. The relatively narrow confidence intervals surrounding the principal mean differences further indicated that the estimated accessibility advantages were reasonably precise within the analyzed sample.

Table 9. Statistical Significance, Effect Sizes, and 95% Confidence Intervals for Principal Accessibility Outcomes

Accessibility Outcome	Accessibility-First Mean (SD)	Conventional Mean (SD)	Mean Difference	95% CI of Difference	p-value	Effect Size
Overall WCAG compliance (%)	91.8 (4.9)	74.6 (8.7)	17.2	13.5 to 20.9	<0.001	d = 2.45
Level A conformance (%)	95.4 (3.5)	82.7 (7.4)	12.7	9.7 to 15.7	<0.001	d = 2.19
Level AA conformance (%)	88.2 (6.1)	68.5 (10.2)	19.7	15.4 to 24.0	<0.001	d = 2.34
Contrast compliance (%)	96.1 (3.2)	79.4 (9.8)	16.7	12.9 to 20.5	<0.001	d = 2.29
Non-text contrast (%)	92.8 (5.1)	72.6 (11.4)	20.2	15.6 to 24.8	<0.001	d = 2.29
Keyboard-accessible components (%)	94.7 (4.6)	76.3 (10.8)	18.4	14.1 to 22.7	<0.001	d = 2.22
Keyboard task success (%)	92.4 (5.8)	70.8 (12.5)	21.6	16.5 to 26.7	<0.001	d = 2.22
Screen-reader compatibility (%)	90.6 (6.2)	69.7 (13.1)	20.9	15.6 to 26.2	<0.001	d = 2.04
Assistive-technology task success (%)	88.9 (7.1)	66.4 (14.2)	22.5	16.7 to 28.3	<0.001	d = 2.01
Labeling errors per platform	4.3 (2.7)	11.8 (5.4)	-7.5	-9.7 to -5.3	<0.001	d = -1.76

Table 9 demonstrated that accessibility-first interfaces achieved statistically significant advantages across all major accessibility outcomes. Effect sizes were consistently large, with absolute Cohen's d values ranging from 1.76 for labeling errors to 2.45 for overall WCAG compliance. Level AA conformance, contrast performance, keyboard accessibility, and assistive-technology outcomes all demonstrated effects exceeding 2.00 standard deviations. The 95% confidence intervals did not include zero for any reported mean difference, reinforcing the statistical reliability of the observed group effects. These findings showed that the accessibility differences were not merely statistically detectable but were substantial in magnitude and consistently favored accessibility-first development environments.

Confidence Intervals and Precision of Estimated Accessibility Effects

Confidence-interval analysis provided additional evidence regarding the precision and stability of the principal estimates. The 95% confidence interval for the overall WCAG compliance difference indicated that accessibility-first platforms exceeded conventional platforms by approximately 13.5 to 20.9 percentage points. For Level AA conformance, the estimated advantage ranged from 15.4 to 24.0 percentage points, demonstrating that the difference remained substantial even at the lower confidence boundary. Keyboard task success produced an estimated improvement between 16.5 and 26.7 percentage points, while the screen-reader compatibility difference ranged from 15.6 to 26.2 percentage points. The largest absolute functional difference was observed for assistive-technology task success,

where the mean advantage was 22.5 percentage points and the 95% confidence interval extended from 16.7 to 28.3. These intervals indicated that the observed accessibility effects were not dependent on isolated extreme values. The confidence intervals around the regression estimates similarly demonstrated that accessible component reuse and design-token adoption retained meaningful associations with accessibility performance after adjustment for other platform characteristics. Component reuse had a standardized regression coefficient of -0.46 for normalized WCAG violations, whereas design-token adoption produced a standardized coefficient of 0.52 for contrast compliance. Both effects remained statistically significant at $p < 0.001$, confirming their relative importance within the multivariable models.

Effect Sizes and Explanatory Strength of Design-System Predictors

The regression and multigroup analyses further demonstrated that accessibility-oriented design-system characteristics accounted for substantial variation in enterprise accessibility outcomes. The multiple regression model predicting WCAG violations per 100 interface elements produced an adjusted R^2 of 0.648, indicating that 64.8% of the observed variation in normalized violation frequency was explained by the included predictors. Accessible component reuse produced the strongest standardized effect within this model at $\beta = -0.46$, followed by accessibility testing coverage at $\beta = -0.27$ and interface complexity at $\beta = 0.24$. The model predicting WCAG contrast compliance explained 67.2% of the variance, with accessibility-oriented design-token adoption emerging as the strongest predictor at $\beta = 0.52$. The contrast-failure model explained 61.9% of variance and showed that greater token adoption was associated with fewer contrast defects, while local styling overrides were associated with increased failure frequency. Multigroup analysis produced similarly large effects. Design-system maturity generated $\eta^2 = 0.595$, accessible component reuse intensity produced $\eta^2 = 0.563$, and design-token adoption generated $\eta^2 = 0.608$. These values indicated that accessibility performance differed substantially across levels of design-system implementation and that the strongest subgroup effect was associated with accessibility-oriented token adoption.

Table 10. Integrated Effect Sizes and Explanatory Statistics for Accessibility Predictors

Analysis / Predictor	Outcome Variable	Effect Statistic	95% CI	p-value	Variance / Effect Magnitude
Accessible component reuse	Violations per 100 elements	$\beta = -0.46$	-0.61 to -0.29	<0.001	Adjusted $R^2 = 0.648$
Accessibility testing coverage	Violations per 100 elements	$\beta = -0.27$	-0.43 to -0.08	0.006	Model predictor
Interface complexity	Violations per 100 elements	$\beta = 0.24$	0.05 to 0.42	0.013	Model predictor
Design-token adoption	Contrast compliance	$\beta = 0.52$	0.34 to 0.68	<0.001	Adjusted $R^2 = 0.672$
Component reuse	Contrast compliance	$\beta = 0.23$	0.05 to 0.40	0.014	Model predictor
Local styling overrides	Contrast compliance	$\beta = -0.29$	-0.46 to -0.10	0.003	Model predictor
Design-token adoption	Contrast failures	$\beta = -0.49$	-0.65 to -0.29	<0.001	Adjusted $R^2 = 0.619$
Design-system maturity	Overall WCAG compliance	$\eta^2 = 0.595$	0.41 to 0.68	<0.001	Large
Component reuse intensity	Overall WCAG compliance	$\eta^2 = 0.563$	0.38 to 0.66	<0.001	Large
Design-token adoption level	Contrast compliance	$\eta^2 = 0.608$	0.43 to 0.70	<0.001	Large
Interface complexity	Overall accessibility outcomes	$\eta^2 = 0.194$	0.05 to 0.31	0.002	Moderate-to-large

Table 10 showed that accessibility-oriented design-system characteristics explained substantial variation in the measured outcomes. Component reuse remained the strongest predictor of normalized WCAG violations, while design-token adoption produced the largest standardized effect on contrast compliance. The corresponding regression models explained between 61.9% and 67.2% of outcome variance. Multigroup effects were also large, particularly for token adoption, design-system maturity, and component reuse intensity. Interface complexity contributed significantly to accessibility variation but produced a smaller effect than the principal accessibility-oriented design practices. Collectively, the confidence intervals and effect estimates confirmed the statistical and practical importance of standardized accessibility infrastructure.

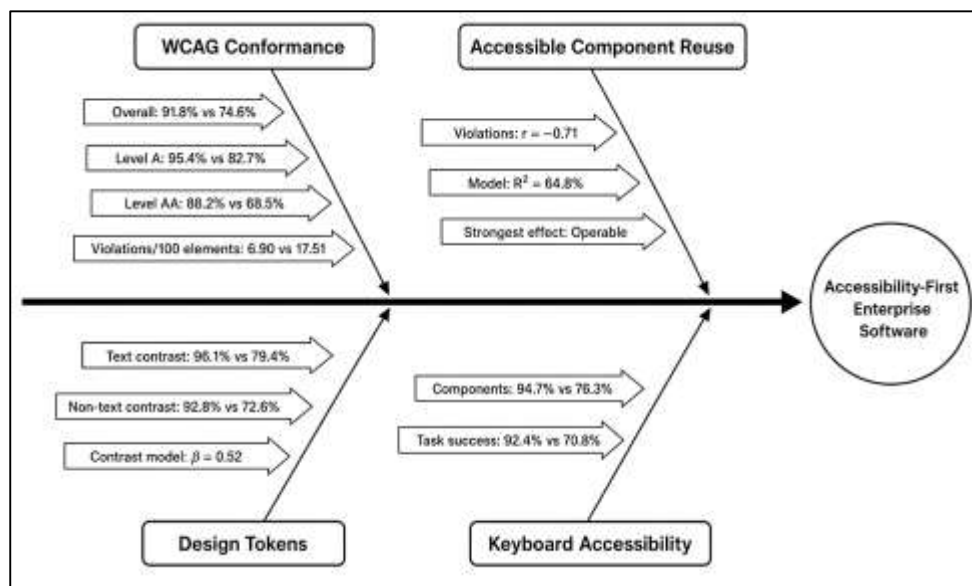
DISCUSSION

The findings of this study demonstrated a substantial difference in WCAG 2.1 accessibility performance between accessibility-first and conventional enterprise software environments. Accessibility-first platforms achieved an overall WCAG 2.1 compliance rate of 91.8%, compared with 74.6% for conventional platforms, while Level A conformance reached 95.4% and 82.7%, respectively. The difference became more pronounced for Level AA, where accessibility-first platforms achieved 88.2% conformance compared with 68.5% among conventional platforms. This pattern was consistent with earlier studies of web applications, institutional platforms, commercial systems, and complex digital interfaces, which generally reported that foundational accessibility requirements were more frequently satisfied than higher-level requirements (Agrawal et al., 2022). Earlier research has also demonstrated that Level AA criteria often expose persistent problems involving contrast, focus visibility, navigation consistency, error identification, and complex interface interaction. The present findings extended this pattern to enterprise software and indicated that accessibility-first development was associated with a substantially smaller deterioration between Level A and Level AA performance. Importantly, the results also confirmed the limitations of interpreting accessibility exclusively through overall conformance percentages. Conventional platforms accounted for 70.6% of all detected violations despite representing only half of the evaluated platforms, and their normalized violation frequency reached 17.51 violations per 100 interface elements compared with 6.90 among accessibility-first platforms (Agrawal et al., 2019). Earlier accessibility studies similarly emphasized that compliance percentages can obscure the repeated occurrence of individual defects, particularly where a single inaccessible component or template generates the same problem across many screens. The current results strengthened this interpretation by demonstrating simultaneous differences in compliance, violation density, and severity. Critical and serious violations were proportionally more common among conventional platforms, indicating that accessibility-first environments were associated not merely with fewer failures but also with a lower concentration of high-impact defects. The very large effect observed for overall WCAG compliance further indicated that the difference was practically meaningful rather than only statistically detectable (Campoverde-Molina et al., 2021). Consequently, the findings supported earlier arguments that accessibility maturity is better represented through multiple complementary indicators, including standards conformance, normalized violation frequency, error density, and defect severity. Within enterprise environments, accessibility-first development appeared to establish a more systematic technical foundation for achieving WCAG 2.1 requirements across complex and repeated interface structures.

The distribution of accessibility failures across Perceivable, Operable, Understandable, and Robust requirements provided additional insight into the nature of enterprise accessibility problems. Perceivable violations represented the largest category in the complete dataset, accounting for approximately 40% of detected violations, followed by Operable violations at approximately 27%. Robust and Understandable requirements represented smaller proportions, although both remained important sources of functional accessibility barriers (Acosta-Vargas et al., 2018). This pattern was broadly consistent with earlier accessibility evaluations that identified contrast deficiencies, missing or inadequate text alternatives, structural presentation problems, keyboard barriers, and focus-management failures among the most common problems in digital interfaces. Previous research has frequently reported particularly high levels of Perceivable failures because visual properties and content structures are repeated extensively across web pages and application screens. The present findings showed that this pattern remained evident in enterprise environments characterized by

dashboards, forms, tables, navigation structures, administrative workflows, and interactive controls. However, accessibility-first platforms recorded fewer violations in every POUR category, suggesting that their advantage was not restricted to one accessibility principle.

Figure 12: Accessibility-first enterprise software framework



The strongest relative differences were observed in Perceivable and Operable accessibility, which corresponded closely with areas that can be standardized through shared visual tokens, reusable components, keyboard interaction patterns, and focus behavior (Quintal & Macias, 2021). Earlier design-system literature has proposed that standardized components are particularly effective for requirements that can be encoded directly into interface infrastructure. The current findings provided quantitative support for this proposition. Understandable accessibility demonstrated a smaller association with design-system practices, which was also consistent with previous studies emphasizing that comprehensibility depends heavily on application-specific language, instructions, workflow organization, error communication, and contextual predictability. Robust accessibility showed a meaningful improvement among accessibility-first interfaces, supporting earlier evidence that reusable semantic structures and validated ARIA implementation can improve assistive-technology compatibility. At the same time, the persistence of Robust violations indicated that standardized components did not completely eliminate application-level semantic problems (Ali et al., 2017). The findings therefore reinforced the argument that POUR dimensions should not be treated as equally responsive to technical standardization. Perceivable, Operable, and parts of Robust accessibility appeared particularly responsive to design-system infrastructure, whereas Understandable accessibility remained more dependent on contextual design and content decisions. This differential pattern explains why aggregate WCAG scores alone provide an incomplete account of enterprise accessibility quality.

One of the most important findings of this study concerned the statistical relationship between accessible component reuse and WCAG 2.1 performance. Component reuse was strongly negatively associated with violations per 100 interface elements and accessibility error density, while strong positive relationships were identified with overall WCAG compliance, Level AA conformance, keyboard-accessible component rates, and keyboard task success (Alajarmeh, 2022). The correlation between component reuse and normalized violation frequency reached -0.71 , and component reuse remained a significant predictor in the multivariable model after interface complexity, testing coverage, and other relevant characteristics were considered. The model explained 64.8% of the variance in normalized violation frequency, demonstrating substantial explanatory strength. These findings were consistent with earlier component-based development and accessibility literature, which characterized reusable accessible components as a means of embedding semantic structure, keyboard functionality,

labeling conventions, focus behavior, and interaction patterns into shared technical infrastructure (Katerattanakul et al., 2018). Previous studies have repeatedly observed that accessibility defects tend to recur when common interface controls are independently recreated across pages or applications. The present results suggested that extensive reuse of validated components substantially reduced this repetition. The strongest POUR-specific association occurred for Operable accessibility, supporting earlier arguments that reusable components are especially effective for standardizing keyboard interaction, focus management, control behavior, and navigation mechanics. Nevertheless, the findings also supported previous warnings that component reuse should not be interpreted as an automatic guarantee of WCAG conformance. The association with Understandable violations was weaker, indicating that reusable infrastructure had less influence on accessibility requirements governed by content, terminology, instructions, and application-specific workflows. Furthermore, the statistical relationship depended on the accessibility quality of the reused components (Altinier et al., 2022). Earlier design-system guidance has emphasized that defective shared components can propagate accessibility failures at scale just as effectively as accessible components can propagate correct behavior. The present findings were compatible with this dual interpretation. Component reuse appeared beneficial when accessibility was embedded within reusable infrastructure, but reuse itself was not sufficient without component validation and appropriate application-level implementation. This study therefore extended earlier conceptual claims by demonstrating quantitatively that accessible component reuse was one of the strongest predictors of lower enterprise accessibility defect frequency while remaining conditional on implementation quality and accessibility governance (Al Jabor et al., 2021).

The findings concerning accessibility-oriented design tokens provided strong evidence that centralized visual standards were associated with improved contrast accessibility. Accessibility-first platforms achieved a mean text-background contrast compliance rate of 96.1%, compared with 79.4% for conventional platforms, while non-text contrast conformance reached 92.8% and 72.6%, respectively. Design-token adoption showed a strong positive correlation with overall contrast compliance and a strong negative relationship with contrast failures per interface. Within the multivariable analysis, accessibility-oriented token adoption remained the strongest predictor of contrast compliance, with a standardized coefficient of 0.52, while the overall model explained 67.2% of variance (Singh et al., 2021). These results were consistent with earlier studies that repeatedly identified insufficient contrast as one of the most common accessibility problems across websites, applications, public-sector platforms, educational systems, and commercial digital environments. Previous literature has suggested that decentralized color selection contributes to this problem because local styling allows designers and developers to introduce untested foreground-background combinations, inconsistent component states, and branding variations that do not consistently satisfy WCAG requirements. The current findings supported this explanation by showing that local styling overrides were negatively associated with contrast compliance and positively associated with contrast failures. Accessibility-oriented design tokens appeared to reduce this variability by establishing reusable visual decisions at the system level (Dias et al., 2022). The results were also consistent with earlier design-system research emphasizing the importance of semantic color roles rather than unrestricted color values. Centralized definitions for text, backgrounds, borders, icons, focus indicators, error states, and selected states can reduce the number of accessibility decisions that must be independently repeated during application development. However, the findings also indicated that text contrast responded more strongly to token adoption than some forms of non-text contrast. This difference was compatible with previous research showing that non-text accessibility depends on how tokens are applied to control boundaries, icons, graphical objects, focus states, and interaction states within particular components. Therefore, token availability alone did not ensure correct usage (Xu et al., 2016). Overall, the findings extended previous accessibility research by demonstrating that centralized accessible visual infrastructure was associated with materially stronger contrast outcomes and that uncontrolled local styling represented an important source of visual accessibility risk within enterprise software.

Keyboard accessibility findings demonstrated that accessibility-first development was associated with both stronger technical operability and improved task-level performance. Accessibility-first platforms achieved a keyboard-accessible component rate of 94.7%, compared with 76.3% for conventional

platforms, while keyboard task success reached 92.4% and 70.8%, respectively. Conventional platforms also produced substantially more navigation errors, focus-order problems, and keyboard traps. These findings were consistent with earlier accessibility and human-computer interaction studies that identified keyboard operability as one of the most important determinants of functional accessibility in complex digital systems (Godinho et al., 2015). Previous research has demonstrated that technically reachable controls do not necessarily create an efficient keyboard experience because logical tab sequences, visible focus, predictable navigation, and appropriate focus transitions are also necessary for successful interaction. The current findings strongly supported this distinction. Accessibility-first interfaces did not merely contain a greater proportion of keyboard-operable components; users were also able to complete keyboard-based workflows more successfully. This relationship was especially important in the enterprise context because business applications frequently require repeated interaction with forms, dialogs, tables, menus, filters, dashboards, and administrative processes. Small navigation inefficiencies can accumulate across repeated tasks, while severe defects such as keyboard traps can prevent task completion entirely. Earlier studies have also emphasized that custom widgets frequently introduce accessibility barriers when native interaction expectations are replaced by poorly implemented keyboard behavior (Khan & Khusro, 2022). The lower frequency of keyboard defects in accessibility-first platforms suggested that reusable and tested component patterns reduced this risk. The large effect sizes associated with keyboard-accessible components and keyboard task success further demonstrated that the differences were practically substantial. The findings also corresponded with the strong relationship between component reuse and Operable accessibility, suggesting that keyboard performance was one of the areas most directly influenced by standardized component infrastructure. At the same time, the remaining keyboard errors among accessibility-first platforms demonstrated that design-system adoption did not completely eliminate application-level failures. Context-specific focus movement, dynamic content, and workflow composition continued to require direct evaluation (Babaian et al., 2016). Thus, the present findings reinforced earlier recommendations that keyboard accessibility should be assessed through both technical component compliance and realistic end-to-end task performance.

The screen-reader and assistive-technology findings further demonstrated that the accessibility advantage of accessibility-first interfaces extended beyond automated WCAG checks into functional interaction. Accessibility-first platforms achieved a mean screen-reader compatibility score of 90.6%, compared with 69.7% for conventional platforms, representing a difference of 20.9 percentage points and a very large standardized effect. Assistive-technology task success showed an even larger absolute difference, reaching 88.9% among accessibility-first platforms and 66.4% among conventional platforms (Buzzi et al., 2019). These findings were consistent with earlier research emphasizing that automated accessibility testing cannot fully determine whether an interface functions effectively with screen readers. Previous studies have demonstrated that technically correct markup may still produce poor experiences when reading order is inappropriate, focus transitions are confusing, dynamic updates are not announced, accessible names are ambiguous, or ARIA states do not accurately represent component behavior. The current findings supported the importance of combining technical conformance measures with realistic assistive-technology task testing. Conventional platforms contained more ARIA-related problems and inaccessible form controls, which corresponded with their lower screen-reader compatibility and task success. This pattern aligned with earlier evidence that incorrect semantic implementation can create substantial barriers even when an interface appears visually complete. The stronger performance of accessibility-first systems also supported design-system guidance that recommends testing reusable components using multiple assistive technologies, browsers, and interaction methods before broad deployment (Senjam et al., 2021). Standardized components appeared to improve the consistency of roles, states, accessible names, and interaction behavior, thereby reducing repeated compatibility failures. However, the gap between screen-reader compatibility and perfect performance remained notable even among accessibility-first platforms. Earlier studies have similarly shown that component-level accessibility can deteriorate when application teams modify component content, override semantics, or combine individually accessible components into inaccessible workflows. Therefore, the present findings did not suggest that design systems replaced assistive-technology testing. Instead, they indicated that accessibility-first

infrastructure provided a stronger baseline from which application-level compatibility could be achieved (Petrovčič et al., 2018). The findings consequently reinforced the importance of evaluating enterprise accessibility through both standards-based technical measures and actual task completion with assistive technologies.

The integrated statistical findings provided strong evidence that accessibility-first enterprise development was associated with broad and practically meaningful improvements across WCAG conformance, violation frequency, visual accessibility, keyboard interaction, and assistive-technology performance. The principal group comparisons were statistically significant, but the interpretation was strengthened considerably by the magnitude of the observed effects (Zdravkova, 2022). Cohen's *d* values exceeded 2.00 for several major outcomes, including overall WCAG compliance, Level AA conformance, contrast accessibility, keyboard task success, and screen-reader compatibility. Confidence intervals for the major group differences remained well separated from zero, indicating that the observed advantages were not dependent on marginal estimates. Multigroup analyses produced similarly large effects for design-system maturity, component reuse intensity, and accessibility-oriented token adoption. These findings were consistent with earlier studies arguing that accessibility is most effective when incorporated throughout design and development rather than treated as a final-stage remediation activity. Previous literature has repeatedly identified fragmented responsibility, inconsistent component implementation, locally defined styling, limited assistive-technology testing, and reactive accessibility correction as contributors to recurring accessibility failures (Cistola et al., 2021). The current study extended these arguments by demonstrating that higher design-system maturity was associated with progressively stronger quantitative accessibility performance. High-maturity platforms achieved markedly greater WCAG compliance and lower violation frequency than low-maturity platforms, while high component reuse corresponded with stronger keyboard performance and high token adoption corresponded with substantially better contrast compliance. Interface complexity remained a significant risk factor, but its effect was smaller than those associated with the principal accessibility-oriented design-system characteristics. This result was particularly important because it suggested that complexity alone did not explain poor accessibility outcomes; rather, the manner in which complexity was managed through standardized accessible infrastructure was strongly associated with performance (Tveter et al., 2017). The findings therefore supported a systems-oriented interpretation of enterprise accessibility in which WCAG conformance emerged from the interaction of design-system maturity, component quality, token governance, testing coverage, and application-level implementation. In comparison with earlier studies that often examined isolated websites, individual accessibility defects, or specific testing methods, this study provided an integrated quantitative perspective linking accessibility-first development practices with multiple technical and functional outcomes (Li et al., 2022). Overall, the convergence of compliance measures, violation metrics, effect sizes, regression results, keyboard performance, and assistive-technology outcomes indicated that accessibility-first design systems represented a substantial organizational and technical mechanism for improving accessibility across enterprise software ecosystems (El Maissi et al., 2022).

CONCLUSION

This study concluded that accessibility-first design-system practices were strongly associated with improved WCAG 2.1 conformance and substantially better functional accessibility across enterprise software interfaces. The quantitative findings demonstrated that accessibility-first platforms consistently outperformed conventional platforms in overall WCAG compliance, Level A and Level AA conformance, violation frequency, accessibility error density, contrast compliance, keyboard operability, screen-reader compatibility, and assistive-technology task success. The distribution of accessibility failures across the POUR dimensions further showed that Perceivable and Operable requirements accounted for the greatest proportions of detected violations, while Understandable and Robust failures occurred less frequently but remained functionally important. The results demonstrated that accessible component reuse represented a particularly influential design-system characteristic, as higher reuse rates were associated with fewer WCAG violations, lower accessibility error density, stronger keyboard accessibility, and higher overall conformance. Accessibility-oriented design-token adoption also emerged as an important determinant of visual accessibility by supporting

more consistent text-background combinations, stronger non-text contrast, and fewer repeated contrast failures. The comparative findings indicated that accessibility-first platforms experienced substantially fewer keyboard defects, labeling errors, ARIA problems, inaccessible form controls, and serious accessibility violations than conventional platforms. These differences were supported not only by statistical significance but also by large effect sizes and confidence intervals that demonstrated the practical importance and precision of the observed outcomes. Regression analyses further established that component reuse, design-token adoption, accessibility testing coverage, and design-system maturity explained substantial variation in accessibility performance even when interface complexity and other platform characteristics were considered. At the same time, the findings demonstrated that reusable components and standardized tokens did not automatically guarantee complete accessibility because application-level configuration, local styling overrides, semantic implementation, content quality, and workflow design continued to influence conformance. Overall, the study established that enterprise accessibility was most effectively understood as a system-level quality outcome rather than an isolated compliance activity. Embedding accessibility into reusable components, design tokens, interaction standards, testing procedures, and design-system governance was associated with more consistent and scalable accessibility outcomes. The convergence of WCAG measures, POUR-specific results, violation severity, keyboard performance, contrast outcomes, and assistive-technology task success demonstrated that accessibility-first design systems provided a stronger technical foundation for creating enterprise software that was more perceivable, operable, understandable, robust, and functionally accessible across diverse interaction requirements.

RECOMMENDATION

Based on the findings of this study, enterprise software organizations should adopt accessibility-first design systems as a core component of software development, interface governance, and quality assurance rather than treating accessibility as a separate activity performed primarily during final-stage compliance testing. Organizations should prioritize the development and systematic reuse of accessible components for frequently occurring interface elements, including buttons, forms, navigation menus, dialogs, tabs, tables, dropdowns, alerts, and other interactive controls, because greater accessible component reuse was associated with lower WCAG 2.1 violation frequency and stronger keyboard accessibility. Each shared component should incorporate validated semantic structure, accessible naming, keyboard behavior, focus management, and assistive-technology support before being released for application-level use. Accessibility-oriented design tokens should also be established for foreground and background colors, typography, borders, icons, focus indicators, validation states, and other visual properties to reduce locally introduced contrast failures and promote consistent visual accessibility across products. Local styling overrides should be controlled through documented governance procedures because uncontrolled customization can weaken the accessibility benefits provided by centralized tokens and components. Enterprise teams should evaluate accessibility through multiple measures rather than relying exclusively on overall WCAG compliance percentages. Level A and Level AA conformance, POUR-specific violation rates, accessibility error density, severity classifications, keyboard task success, contrast performance, screen-reader compatibility, and assistive-technology task success should be incorporated into regular accessibility reporting. Automated accessibility testing should be integrated into development and deployment pipelines but should be complemented by manual keyboard evaluation, screen-reader testing, contrast inspection, and realistic end-to-end workflow testing. Particular attention should be directed toward Perceivable and Operable requirements because these dimensions represented the greatest concentrations of violations, while Understandable and Robust requirements should receive equivalent governance attention because lower violation frequency does not necessarily indicate lower functional impact. Accessibility performance should also be monitored according to application complexity, component reuse intensity, token adoption, and design-system maturity to identify areas requiring targeted improvement. Finally, organizations should maintain clear accessibility documentation, component usage requirements, testing protocols, and accountability mechanisms across design, development, content, and quality-assurance teams. Through centralized standards, reusable accessible infrastructure, continuous validation, and application-level testing, enterprise organizations can reduce repeated accessibility defects, improve WCAG 2.1 conformance, strengthen assistive-technology compatibility, and provide

more consistent and efficient interaction across complex software workflows.

LIMITATIONS

This study was subject to several limitations that should be considered when interpreting the quantitative relationships between accessibility-first design-system practices and WCAG 2.1 outcomes across enterprise software interfaces. First, the study employed a cross-sectional comparative design, meaning that accessibility characteristics and outcomes were evaluated within a defined assessment period rather than observed longitudinally throughout the complete design, development, deployment, and maintenance lifecycle. Consequently, the statistically significant relationships identified between accessible component reuse, design-token adoption, design-system maturity, and accessibility performance demonstrated strong associations but did not independently establish causal effects. Second, the sample consisted of selected enterprise platforms and representative workflows, screens, components, and interface elements; therefore, the evaluated sample might not have captured every accessibility condition occurring within larger software ecosystems. Enterprise applications frequently contain role-specific functionality, dynamically generated content, third-party integrations, legacy modules, and infrequently accessed workflows that may present additional accessibility barriers. Third, although automated accessibility testing improved consistency and enabled large numbers of interface elements to be assessed systematically, automated tools could identify only a proportion of WCAG 2.1 failures. Requirements involving meaningful reading order, contextual labeling, understandable instructions, predictable interaction, appropriate alternative text, and some assistive-technology behaviors required manual judgment and remained susceptible to evaluator interpretation. Fourth, screen-reader compatibility and keyboard task performance could vary according to operating system, browser, assistive-technology version, user expertise, device configuration, and interaction strategy, limiting the extent to which the observed task outcomes represented every possible user environment. Fifth, the classification of platforms as accessibility-first or conventional necessarily simplified differences in organizational accessibility maturity because enterprise development environments may incorporate accessibility practices to varying degrees rather than fitting completely into two distinct categories. Component reuse rates and design-token adoption also did not fully capture whether every reused component or token had been implemented correctly within its application context. Sixth, interface complexity, application size, testing coverage, and selected organizational characteristics were considered statistically, but other potential influences, including development resources, release schedules, legacy architecture, third-party dependencies, and accessibility expertise, were not comprehensively controlled. Finally, WCAG 2.1 conformance represented the principal technical accessibility framework, but standards compliance did not necessarily capture every aspect of practical usability, inclusion, efficiency, or user satisfaction. These limitations indicated that the findings were most appropriately interpreted as robust quantitative evidence of relationships between accessibility-first practices and enterprise accessibility performance rather than as universal evidence that design-system adoption alone guaranteed complete accessibility.

REFERENCES

- [1]. Abascal, J., Arrue, M., & Valencia, X. (2019). Tools for web accessibility evaluation. In *Web accessibility: a foundation for research* (pp. 479-503). Springer.
- [2]. Abou-Zahra, S., & Brewer, J. (2019). Standards, guidelines, and trends. In *Web accessibility: A foundation for research* (pp. 225-246). Springer.
- [3]. Acosta-Vargas, P., Acosta, T., & Lujan-Mora, S. (2018). Framework for accessibility evaluation of hospital websites. 2018 International Conference on edemocracy & eGovernment (ICEDEG),
- [4]. Acosta-Vargas, P., Luján-Mora, S., & Salvador-Ullauri, L. (2016). Evaluation of the web accessibility of higher-education websites. 2016 15th International Conference on Information Technology Based Higher Education and Training (ITHET),
- [5]. Acosta-Vargas, P., Luján-Mora, S., & Salvador-Ullauri, L. (2017). Quality evaluation of government websites. 2017 Fourth International Conference on eDemocracy & eGovernment (ICEDEG),
- [6]. Acosta-Vargas, P., Salvador-Acosta, B., Salvador-Ullauri, L., Villegas-Ch, W., & Gonzalez, M. (2021). Accessibility in native mobile applications for users with disabilities: A scoping review. *Applied Sciences*, 11(12), 5707.
- [7]. Acosta-Vargas, P., Salvador-Ullauri, L. A., & Luján-Mora, S. (2019). A heuristic method to evaluate web accessibility for users with low vision. *IEEE access*, 7, 125634-125648.
- [8]. Afonso, A. P., Pinto, A. S., Braga, I., & Oliveira, R. (2022). E-Commerce Website Accessibility Assessment Portugal Case Overview. In *Advances in Tourism, Technology and Systems: Selected Papers from ICOTTS 2021, Volume 2* (pp. 491-500). Springer.

- [9]. Agrawal, G., Kumar, D., & Singh, M. (2022). Assessing the usability, accessibility, and mobile readiness of e-government websites: a case study in india. *Universal Access in the Information Society*, 21(3), 737-748.
- [10]. Agrawal, G., Kumar, D., Singh, M., & Dani, D. (2019). Evaluating accessibility and usability of airline websites. *International conference on advances in computing and data sciences*,
- [11]. Akgül, Y. (2017). The most violated WCAG 1.0 guidelines by the developers of university websites in Turkey. 2017 12th Iberian Conference on Information Systems and Technologies (CISTI),
- [12]. Al Jabor, A. N., Adnan, F., Park, M., & Othman, A. (2021). Mada web accessibility monitor tool. 2021 8th International Conference on ICT & Accessibility (ICTA),
- [13]. Alajarmeh, N. (2022). The extent of mobile accessibility coverage in WCAG 2.1: sufficiency of success criteria and appropriateness of relevant conformance levels pertaining to accessibility problems encountered by users who are visually impaired. *Universal Access in the Information Society*, 21(2), 507-532.
- [14]. Ali, S., AlBalushi, T., & AlBadi, A. (2017). Guidelines and deployment of accessibility-aware framework approach. *International Journal of Web Information Systems*, 13(2), 114-139.
- [15]. Alsaedi, A. (2020). Comparing web accessibility evaluation tools and evaluating the accessibility of webpages: proposed frameworks. *Information*, 11(1), 40.
- [16]. Altinier, A., Oncins, E., Sauberer, G., & Mehigan, T. (2022). Demystifying digital accessibility and fostering inclusive mindsets. Compliance with the European standard for digital accessibility EN 301 549. *European Conference on Software Process Improvement*,
- [17]. Babaian, T., Lucas, W., Chircu, A. M., & Power, N. (2016). Interactive visualizations for workplace tasks. *International Conference on Enterprise Information Systems*,
- [18]. Baptista, A., Martins, J., Goncalves, R., Branco, F., & Rocha, T. (2016). Web accessibility challenges and perspectives: A systematic literature review. 2016 11th Iberian Conference on Information Systems and Technologies (CISTI),
- [19]. Berardinucci, F., Colombo, G., Lorusso, M., Manzini, M., Terkaj, W., & Urgo, M. (2022). A learning workflow based on an integrated digital toolkit to support education in manufacturing system engineering. *Journal of Manufacturing Systems*, 63, 411-423.
- [20]. Bonneau, J., Miller, A., Clark, J., Narayanan, A., Kroll, J. A., & Felten, E. W. (2015). Sok: Research perspectives and challenges for bitcoin and cryptocurrencies. 2015 IEEE symposium on security and privacy,
- [21]. Brajnik, G., & Vigo, M. (2019). Automatic web accessibility metrics: Where we were and where we went. In *Web Accessibility: A Foundation for Research* (pp. 505-521). Springer.
- [22]. Buzzi, M., Leporini, B., & Meattini, C. (2019). Design guidelines for web interfaces of home automation systems accessible via screen reader. *Journal of Web Engineering*, 18(4-6), 477-511.
- [23]. Campoverde-Molina, M., Lujan-Mora, S., & Garcia, L. V. (2020). Empirical studies on web accessibility of educational websites: A systematic literature review. *IEEE access*, 8, 91676-91700.
- [24]. Campoverde-Molina, M., Luján-Mora, S., & Valverde, L. (2021). Process model for continuous testing of web accessibility. *IEEE access*, 9, 139576-139593.
- [25]. Chadha, S. (2022). *Beyond Accessibility Compliance*. Springer.
- [26]. Chakraborti, T., Isahagian, V., Khalaf, R., Khazaeni, Y., Muthusamy, V., Rizk, Y., & Unuvar, M. (2020). From Robotic Process Automation to Intelligent Process Automation: -Emerging Trends-. *International Conference on Business Process Management*,
- [27]. Chee, M., Davidian, Z., & Weaver, K. D. (2022). More to do than can ever be done: Reconciling library online learning objects with WCAG 2.1 standards for accessibility. *Journal of Web Librarianship*, 16(2), 87-119.
- [28]. Chee, M., & Weaver, K. D. (2021). Meeting a higher standard: A case study of accessibility compliance in LibGuides upon the adoption of WCAG 2.0 guidelines. *Journal of Web Librarianship*, 15(2), 69-89.
- [29]. Cistola, G., Farrús, M., & van der Meulen, I. (2021). Aphasia and acquired reading impairments: What are the high-tech alternatives to compensate for reading deficits? *International Journal of Language & Communication Disorders*, 56(1), 161-173.
- [30]. Debasish, A. (2021). Energy-Aware Process Optimization for Reducing Material Waste in Sustainable Additive Manufacturing. *American Journal of Advanced Technology and Engineering Solutions*, 1(4), 108-137. <https://doi.org/10.63125/j0wk7j48>
- [31]. Dias, J., Carvalho, D., Rocha, T., & Barroso, J. (2022). Automated Evaluation Tools for Web and Mobile Accessibility: proposal of a new adaptive interface tool. *Procedia Computer Science*, 204, 297-304.
- [32]. Domínguez Vila, T., Alén González, E., & Darcy, S. (2018). Website accessibility in the tourism industry: an analysis of official national tourism organization websites around the world. *Disability and rehabilitation*, 40(24), 2895-2906.
- [33]. El Maissi, A. M., Argyroudis, S. A., Kassem, M. M., Leong, L. V., & Mohamed Nazri, F. (2022). [Retracted] Integrating Intrinsic and Eccentric Seismic Vulnerability Indices to Prioritize Road Network Accessibility. *Advances in Civil Engineering*, 2022(1), 5888020.
- [34]. Ferati, M., & Vogel, B. (2020). Accessibility in web development courses: A case study. *Informatics*,
- [35]. Fernández-Díaz, E., Correia, M. B., & Matos, N. d. (2021). Portuguese and Spanish DMOs' accessibility apps and websites. *Journal of Theoretical and Applied Electronic Commerce Research*, 16(4), 874-899.
- [36]. Ferri, D., & Favalli, S. (2018). Web accessibility for people with disabilities in the European Union: Paving the road to social inclusion. *Societies*, 8(2), 40.
- [37]. Fröwis, M., Fuchs, A., & Böhme, R. (2019). Detecting token systems on ethereum. *International conference on financial cryptography and data security*,
- [38]. Gaggi, O., & Pederiva, V. (2021). WCAG4All, a tool for making web accessibility rules accessible. 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC),

- [39]. Gambino, O., Pirrone, R., & Giorgio, F. D. (2016). Accessibility of the Italian institutional web pages: a survey on the compliance of the Italian public administration web pages to the Stanca Act and its 22 technical requirements for web accessibility. *Universal Access in the Information Society*, 15(2), 305-312.
- [40]. Garcés-Freire, E., Pailiacho-Mena, V., & Chucuri-Yachimba, J. (2022). Evaluation of web accessibility in the state citizen service portals most used by disabled people. *International Conference on Computer Science, Electronics and Industrial Engineering (CSEI)*,
- [41]. García-Santiago, L., & Olvera-Lobo, M.-D. (2021). How accessibility guidelines are used in Spanish World Heritage websites: an exploratory study. *Library Hi Tech*, 39(1), 144-165.
- [42]. Ghaemmaghani, M., Hanley, G. P., & Jessel, J. (2016). Contingencies promote delay tolerance. *Journal of Applied Behavior Analysis*, 49(3), 548-575.
- [43]. Godinho, R., Condado, P. A., Zacarias, M., & Lobo, F. G. (2015). Improving accessibility of mobile devices with EasyWrite. *Behaviour & Information Technology*, 34(2), 135-150.
- [44]. Gramazio, C. C., Laidlaw, D. H., & Schloss, K. B. (2016). Colorgorical: Creating discriminable and preferable color palettes for information visualization. *IEEE transactions on visualization and computer graphics*, 23(1), 521-530.
- [45]. Haarmann, S., Batoulis, K., & Weske, M. (2018). Compliance checking for decision-aware process models. *International Conference on Business Process Management*,
- [46]. Haße, H., van der Valk, H., Möller, F., & Otto, B. (2022). Design Principles for Shared Digital Twins in Distributed Systems: Design Principles for Shared Digital Twins in Distributed Systems. *Business & information systems engineering*, 64(6), 751-772.
- [47]. Hong, S. G., Trimi, S., Kim, D. W., & Hyun, J. H. (2015). A Delphi study of factors hindering web accessibility for persons with disabilities. *Journal of Computer Information Systems*, 55(4), 28-34.
- [48]. Hossain, M. A., Dwivedi, Y. K., & Naseem, S. B. (2015). Developing and validating a hierarchical model of service quality of retail banks. *Total Quality Management & Business Excellence*, 26(5-6), 534-549.
- [49]. Hu, Y., & Chen, Y. (2022). Coupling of urban economic development and transportation system: An urban agglomeration case. *Sustainability*, 14(7), 3808.
- [50]. Ismail, A., & Kuppusamy, K. (2022). Web accessibility investigation and identification of major issues of higher education websites with statistical measures: A case study of college websites. *Journal of King Saud University Computer and Information Sciences*, 34(3), 901-911.
- [51]. Kaplan, N., & Omer, I. (2022). Multiscale accessibility – a new perspective of space structuration. *Sustainability*, 14(9), 5119.
- [52]. Kashif, S., Ahmad, M., Shahid, M., Habib, M. A., & Rizwan, K. (2022). Development of an automated accessibility evaluation plugin tool for mobile applications. *2022 3rd international conference on innovations in computer science & software engineering (ICONICS)*,
- [53]. Katerattanakul, P., Hong, S., Lee, H.-M., & Kam, H.-J. (2018). The effects of web accessibility certification on the perception of companies' corporate social responsibility. *Universal Access in the Information Society*, 17(1), 161-173.
- [54]. Kazuye Kimura, A. (2018). Defining, evaluating, and achieving accessible library resources: A review of theories and methods. *Reference Services Review*, 46(3), 425-438.
- [55]. Khan, A., & Khuroso, S. (2022). A mechanism for blind-friendly user interface adaptation of mobile apps: A case study for improving the user experience of the blind people. *Journal of Ambient Intelligence and Humanized Computing*, 13(5), 2841-2871.
- [56]. Kim, M. J., Schroeder, S., Chan, S., Hickerson, K., & Lee, Y.-C. (2022). Reviewing the user-centered design process for a comprehensive Gastroesophageal Reflux Disease (GERD) app. *International Journal of Environmental Research and Public Health*, 19(3), 1128.
- [57]. Kuriyama, S., Mukai, T., Taketomi, T., & Mukasa, T. (2022). Context-based style transfer of tokenized gestures. *Computer Graphics Forum*,
- [58]. Kwak, A., & Newman, J. (2018). An accessibility-first approach to online course readers. *Reference Services Review*, 46(3), 340-349.
- [59]. Leitner, M.-L., Strauss, C., & Stummer, C. (2016). Web accessibility implementation in private sector organizations: motivations and business impact. *Universal Access in the Information Society*, 15(2), 249-260.
- [60]. Leria, L. A., Benitez, P., & Fraga, F. J. (2021). Assistive technology in large-scale assessments for students with visual impairments: A systematic review and recommendations based on the Brazilian reality. *Education and Information Technologies*, 26(3), 3543-3573.
- [61]. Li, G., Xiong, Z., & Song, Y. (2022). A systematic review of quantitative measurement methods for accessibility of urban infrastructure. *International Symposium on Advancement of Construction Management and Real Estate*,
- [62]. Liu, X., Xu, M., Teng, T., Huang, G., & Mei, H. (2016). MUIT: a domain-specific language and its middleware for adaptive mobile web-based user interfaces in WS-BPEL. *IEEE Transactions on Services Computing*, 12(6), 955-969.
- [63]. Martins, J., Gonçalves, R., & Branco, F. (2017). A full scope web accessibility evaluation procedure proposal based on Iberian eHealth accessibility compliance. *Computers in Human Behavior*, 73, 676-684.
- [64]. Md Asif Ali Sheak, A., & Risha, A. (2022). The Role of IOT-Driven Energy Analytics in Improving Operational Efficiency and Sustainability in Modern Industrial Facilities. *American Journal of Interdisciplinary Studies*, 3(04), 807-841. <https://doi.org/10.63125/gm63q956>
- [65]. Md. Abdur, R., & Iftekhar, A. (2021). Customer Retention Forecasting in Mobile Wallet Services Using Neural Networks: A Comparative Quantitative Study. *International Journal of Business and Economics Insights*, 1(4), 70-102. <https://doi.org/10.63125/dyrpc387>

- [66]. Miranda, D. (2021). A web accessibility requirements framework for agile development. 2021 IEEE 29th International Requirements Engineering Conference (RE),
- [67]. Mobasheri, A., Zipf, A., & Francis, L. (2018). OpenStreetMap data quality enrichment through awareness raising and collective action tools – experiences from a European project. *Geo-spatial Information Science*, 21(3), 234-246.
- [68]. Mohammad Abir Chowdhury, S. (2023). Role of AI-Assisted Risk Assessment in Enhancing Audit Planning Efficiency and Risk Prioritization in Public-Safety Organizations. *International Journal of Scientific Interdisciplinary Research*, 4(4), 529–576. <https://doi.org/10.63125/qd6nn437>
- [69]. Mohammad Abir Chowdhury, S., & Tonay, P. (2022). An Empirical Analysis of the Impact of Robotic Process Automation and Continuous Auditing on the Timeliness And Accuracy Of Financial Reporting Assurance. *International Journal of Business and Economics Insights*, 2(4), 73-126. <https://doi.org/10.63125/7p8dcr47>
- [70]. Mohammad Badrul, A., & Md. Mominul, H. (2023). Machine Learning-Based Assessment of Environmental and Public Health Risks in Sewerage Sludge Management Systems. *American Journal of Advanced Technology and Engineering Solutions*, 3(02), 33-77. <https://doi.org/10.63125/y3yfxa92>
- [71]. Muhammad Zahidul, I. (2023). Comparative Analysis of Machine-Learning Approaches for Pharmaceutical Sales Forecasting and Market-Demand Prediction. *Review of Applied Science and Technology*, 2(02), 38–95. <https://doi.org/10.63125/etcbqf13>
- [72]. Nacheva, R. (2021). Digital inclusion through sustainable web accessibility. International Conference on Digital Transformation and Global Society,
- [73]. Nagaraju, M., Chawla, P., & Rana, A. (2019). A practitioner’s approach to assess the wcag 2.0 website accessibility challenges. 2019 Amity International Conference on Artificial Intelligence (AICAI),
- [74]. Nishat, J., & Mst Kaniz, F. (2022). A Review of the Impact of Covid-19 On Maternal and Child Health Service Utilization in Bangladesh. *American Journal of Scholarly Research and Innovation*, 1(02), 223–269. <https://doi.org/10.63125/vgv5s742>
- [75]. Noh, K.-R., Jeong, E.-S., You, Y.-B., Moon, S.-J., & Kang, M.-B. (2015). A study on the current status and strategies for improvement of web accessibility compliance of public institutions. *Journal of Open Innovation: Technology, Market, and Complexity*, 1(1), 1-17.
- [76]. Pant, M., & Sinha, A. (2020). Evaluating state of web accessibility standards in top airline websites. 2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS),
- [77]. Parajuli, P., & Eika, E. (2020). A comparative study of accessibility and usability of Norwegian university websites for screen reader users based on user experience and automated assessment. International Conference on Human-Computer Interaction,
- [78]. Park, K., & Jung, S. (2022). Designing inclusive websites for people with disabilities as part of an event tourism strategic planning process. *Journal of Convention & Event Tourism*,
- [79]. Paz, F., Paz, F. A., Moquillaza, A., Collantes, L., Arenas, J. J., & Villanueva, D. (2021). Comparison of accessibility tools for the evaluation of e-commerce websites. International Conference on Intelligent Human Systems Integration,
- [80]. Persson, H., Åhman, H., Yngling, A. A., & Gulliksen, J. (2015). Universal design, inclusive design, accessible design, design for all: different concepts – one goal? On the concept of accessibility – historical, methodological and philosophical aspects. *Universal Access in the Information Society*, 14(4), 505-526.
- [81]. Petrovič, A., Rogelj, A., & Dolničar, V. (2018). Smart but not adapted enough: Heuristic evaluation of smartphone launchers with an adapted interface and assistive technologies for older adults. *Computers in Human Behavior*, 79, 123-136.
- [82]. Pinto, A. S., Costa, E., Borges, I., Silva, F., & Abreu, A. (2019). Virtual accessibility on digital business websites and tourist distribution. In *Advances in Tourism, Technology and Smart Systems: Proceedings of ICOTTS 2019* (pp. 93-103). Springer.
- [83]. Quintal, C., & Macias, J. A. (2021). Measuring and improving the quality of development processes based on usability and accessibility. *Universal Access in the Information Society*, 20(2), 203-221.
- [84]. Ribeiro, L. S. F., Bui, T., Collomosse, J., & Ponti, M. (2020). Sketchformer: Transformer-based representation for sketched structure. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),
- [85]. Salvador-Ullauri, L., Acosta-Vargas, P., Gonzalez, M., & Luján-Mora, S. (2020a). Combined method for evaluating accessibility in serious games. *Applied Sciences*, 10(18), 6324.
- [86]. Salvador-Ullauri, L., Acosta-Vargas, P., Gonzalez, M., & Luján-Mora, S. (2020b). A heuristic method for evaluating accessibility in web-based serious games for users with low vision. *Applied Sciences*, 10(24), 8803.
- [87]. Sanchez-Gordon, S., Sánchez-Gordón, M., Yilmaz, M., & O'Connor, R. V. (2019). Integration of accessibility design patterns with the software implementation process of ISO/IEC 29110. *Journal of Software: Evolution and Process*, 31(1), e1987.
- [88]. Scanlon, E., Taylor, Z. W., Raible, J., Bates, J., & Chini, J. J. (2021). Physics webpages create barriers to participation for people with disabilities: five common web accessibility errors and possible solutions. *International Journal of STEM Education*, 8(1), 25.
- [89]. Senjam, S. S., Manna, S., & Bascaran, C. (2021). Smartphones-based assistive technology: accessibility features and apps for people with visual impairment, and its usage, challenges, and usability testing. *Clinical optometry*, 311-322.
- [90]. Shaiful, M., Anisur, R., & Md, A. (2022). A Systematic Literature Review on the Role of Digital Health Twins in Preventive Healthcare For Personal and Corporate Wellbeing. *American Journal of Interdisciplinary Studies*, 3(04), 1-31. <https://doi.org/10.63125/negjw373>
- [91]. Shen, Y., & Zhao, J. (2017). Capacity constrained accessibility of high-speed rail. *Transportation*, 44(2), 395-422.

- [92]. Shen, Z., Zhang, L., Li, R., Hou, J., Liu, C., & Hu, W. (2021). The effects of color combinations, luminance contrast, and area ratio on icon visual search performance. *Displays*, 67, 101999.
- [93]. Shestakevych, T., Pasichnyk, V., Kunanets, N., Medykovskyy, M., & Antonyuk, N. (2018). The content web-accessibility of information and technology support in a complex system of educational and social inclusion. 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT),
- [94]. Sik-Lanyi, C., & Orbán-Mihálykó, É. (2019). Accessibility testing of European health-related websites. *Arabian Journal for Science and Engineering*, 44(11), 9171-9190.
- [95]. Silva, J. S. e., Gonçalves, R., Branco, F., Pereira, A., Au-Yong-Oliveira, M., & Martins, J. (2019). Accessible software development: a conceptual model proposal. *Universal Access in the Information Society*, 18(3), 703-716.
- [96]. Singh, R., Ismail, A., PS, S., & Singh, D. (2021). Compliance of accessibility in tourism websites: a pledge towards disability. *Journal of Hospitality and Tourism Insights*, 4(3), 263-281.
- [97]. Skourlas, C., Tsolakidis, A., Belsis, P., Vassis, D., Kampouraki, A., Kakoulidis, P., & Giannakopoulos, G. A. (2016). Integration of institutional repositories and e-learning platforms for supporting disabled students in the higher education context. *Library Review*, 65(3), 136-159.
- [98]. Spyridonis, F., & Daylamani-Zad, D. (2021). A serious game to improve engagement with web accessibility guidelines. *Behaviour & Information Technology*, 40(6), 578-596.
- [99]. Stephanidis, C. (2021). Design for all in digital technologies. *Handbook of human factors and ergonomics*, 1187-1215.
- [100]. Szafir, D. A. (2017). Modeling color difference for visualization design. *IEEE transactions on visualization and computer graphics*, 24(1), 392-401.
- [101]. Tveter, E., Welde, M., & Odeck, J. (2017). Do fixed links affect settlement patterns: a synthetic control approach. *Research in Transportation Economics*, 63, 59-72.
- [102]. Victor, F., & Lüders, B. K. (2019). Measuring ethereum-based erc20 token networks. International conference on financial cryptography and data security,
- [103]. Wang, J., Su, Y., Chen, Z., Tang, L., Wang, G., & Wang, J. (2022). Assessing the spatial accessibility of urban medical facilities in multi-level and multi-period scales based on web mapping API and an improved potential model. *ISPRS International Journal of Geo-Information*, 11(11), 545.
- [104]. Whitaker, R. (2020). *Developing inclusive mobile apps*. Springer.
- [105]. Worsley, M., & Bar-El, D. (2022). Inclusive Making: designing tools and experiences to promote accessibility and redefine making. *Computer Science Education*, 32(2), 155-187.
- [106]. Xu, S., Cornelio, C., & Gianfortune, M. (2016). Where is Siri? The accessibility design challenges for enterprise touchscreen interfaces. International Conference on HCI in Business, Government, and Organizations,
- [107]. Yao, J., Wencheng, W., Prabhakara, J., Watts-Englert, J., Simmons, I., Mongeon, M., & Tharayil, M. (2017). Crowdsourcing workflow optimization to internal worker crowds. 2017 IEEE International Conference on Cognitive Computing (ICCC),
- [108]. Yousfi, A., De Freitas, A., Dey, A. K., & Saidi, R. (2015). The use of ubiquitous computing for business process improvement. *IEEE Transactions on Services Computing*, 9(4), 621-632.
- [109]. Zdravkova, K. (2022). The potential of artificial intelligence for assistive technology in education. In *Handbook on intelligent techniques in the educational process: Vol 1 recent advances and case studies* (pp. 61-85). Springer.
- [110]. Žuliček, L., Tomić, S., & Bosnić, I. (2021). Adapting modularized web applications to web accessibility standards. 2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO),
- [111]. Zuo, T., Wei, H., & Rohne, A. (2018). Determining transit service coverage by non-motorized accessibility to transit: Case study of applying GPS data in Cincinnati metropolitan area. *Journal of Transport Geography*, 67, 1-11.