<packt>

Inclusive Design for Accessibility

A practical guide to digital accessibility, UX, and inclusive web and app design

Dale Cruse & Denis Boudreau et al.

Foreword by Glenda Sims

Chief Information Accessibility Officer (CIAO) at Deque

Inclusive Design for Accessibility

A practical guide to digital accessibility, UX, and inclusive web and app design

Dale Cruse & Denis Boudreau et al.

<packt>

Inclusive Design for Accessibility

Portfolio Director: Ashwin Nair

Relationship Lead: Nitin Nainani

Project Manager: Aparna Nair

Content Engineer: Hayden Edwards

Technical Editor: Sweety Pagaria

Copy Editor: Safis Editing

Indexer: Rekha Nair

Proofreader: Hayden Edwards

Production Designer: Deepak Chavan

Growth Lead: Sohini Ghosh

First published: August 2025

Production reference: 1140725

Published by Packt Publishing Ltd.
Grosvenor House
11 St Paul's Square
Birmingham
B3 1RB, UK.

ISBN 978-1-83588-822-3
`www.packtpub.com`

To my parents, Paul and Geraldine, who taught me – loudly and without apology – that people with disabilities are just that: people. Your unwavering clarity became my compass. And to my wife, Heather Humble – my fiercest believer, my truest partner. This entire book carries your fingerprints, whether visible or not. You're my favorite.

- Dale Cruse

To those living with invisible disabilities or neurodivergent minds. To those who feel they must hide their truth to stay safe, fit in, or be seen as "enough." To those who've mastered the art of masking, not because they want to, but because they have to. This book is for you. May the world soon be one where we no longer have to mask to belong.

- Denis Boudreau

Foreword

What happens when twelve of the most thoughtful, experienced minds in accessibility come together to write a book? You get something far more powerful than a technical manual – you get a blueprint for designing a digital world that truly works for everyone.

This book is a call to shift our thinking – from designing for disabilities to designing for human diversity. It challenges the idea that accessibility is a narrow specialty. Instead, it presents accessibility as the foundation for innovation, equity, and a better user experience for all.

One line that sticks with me:

"*When we encounter an inaccessible website, we now see a flawed design, not a 'disabled' user.*"

This shift in mindset is game-changing.

These authors – educators, practitioners, researchers, and advocates – invite us into a more expansive, human-centered way of working. They show how ability is fluid, not fixed. How intersectionality complicates assumptions. And how co-creation with disabled users isn't just nice to have – it's essential.

The concepts here are grounded in experience and creatively forward-looking. You'll find solid frameworks such as the Seven Principles of Universal Design, alongside fresh, vibrant approaches such as "co-creation carnivals" and deeply practical advice on how to solve complex accessibility puzzles.

You'll be smarter when you finish this book. But more importantly, you'll be better equipped to help build the kind of digital world we all deserve.

To the twelve brilliant authors: thank you for being my muses. I've had the privilege of working side by side with many of you – learning together, mentoring and being mentored, and most of all, sharing a fierce commitment to digital equality. You inspire me to stretch further. Especially you, Boudreau!

To you – yes, you – reading this: let this book guide you, challenge you, and energize your next step forward. Together, we are the changemakers.

To accessibility and beyond!

Glenda Sims (The good witch of a11y)

Chief Information Accessibility Officer (CIAO) at Deque

Contributors

About the authors

Dale Cruse is a highly sensitive empath driven by a lifelong commitment to equal opportunity, inspired by his parents with disabilities. He created accessibility programs at Twitch and McGraw-Hill Education and led inclusive design initiatives at Microsoft and Deque, where he also helped launch the industry-standard axe DevTools accessibility testing plugin. Dale shared his expertise in the book *Accessibility for Everyone*, and *The Web Ahead* podcast, and is the author of *HTML5 Multimedia Development Cookbook* for Packt Publishing.

Denis Boudreau is an international expert in digital accessibility and inclusive leadership with 25 years of experience in the field, working with leaders and executives who are no longer willing to overlook disability inclusion and want to transform their leadership approach from "inclusive-ish" to truly inclusive. He specializes in supporting organizations wishing to adopt more inclusive leadership and communication practices, particularly for their workforce and clients who live with disabilities, are aging, or are marginalized by technology. Offering thought leaders and communicators the means to extend the reach of their message to a much broader audience, he is also the author of the Amazon bestseller *The Inclusive Speaker*, published in 2023.

About the contributing authors

Dr Angela Young (they/them) is a bold advocate for digital accessibility. They drive change among queer, deaf, disabled, and neurodiverse professionals in tech through education, strategy, and storytelling. A seasoned speaker and 18-year educator, Angela has led accessibility initiatives at scale, empowering thousands to build inclusive digital experiences. They bring humor, heart, and deep expertise to every conversation. Angela holds a doctorate in education and certifications in accessibility and leadership.

Maya Sellon is an inclusive design and accessibility leader with over 20 years of experience across North America, Asia, and Europe. She's currently completing her postgraduate degree in assistive technology and human services at CSUN. Maya has developed and implemented accessibility practices for global organizations, embedding inclusivity at the heart of design and technology. With a career spanning civil engineering, user experience, and digital accessibility, she brings a diverse and well-rounded perspective. A tech enthusiast, Maya believes accessibility is about people – not just rules or technology. In her free time, she enjoys drumming and is training to become a certified solo skydiver.

Julianna Rowsell (she/her) is a chronically ill, neurodivergent, and disabled product leader with ADHD, ankylosing spondylitis, and anemia. They bring 20+ years of expertise in accessibility, disability justice, and equity, challenging extractive systems to co-create inclusive just futures. They hold a master's of design in inclusive design from OCAD University. Julianna has helped organizations such as the Government of Canada and leading tech companies make moves toward big bets that focus on accessible outcomes and equitable futures. In 2022, they founded, with their partner, an accessible, sustainable, regenerative flower farm.

Nandita Gupta is a senior product manager of accessibility at Cisco, where she works on creating accessible and user-friendly products for all. She was named ADCOLOR in Tech 2024, and featured in NASDAQ, Forbes, Times of India, and other media outlets. As a TEDx speaker, she has spoken internationally at numerous conferences and works within the intersection of product and accessibility. She is a board member on different boards within the community including Lime Connect and the Accessibility Leadership Council for Disability: IN. She has been actively involved in various mentoring organizations over the past years and was inducted into the Susan M. Daniels Disability Mentoring Hall of Fame.

Jennifer Chadwick is an award-winning senior digital accessibility strategist and coach, who works in Canada, the UK, Europe, and the US. She teaches organizations to understand the needs of users with disabilities and develops action plans for product design, customer service, marketing, HR, and procurement. Jennifer has spoken at CSUN, AccessU, the University of Toronto, and the United Nations COSP. She co-chairs the W3C Accessibility Roles and Responsibilities Mapping (ARRM) community group and contributed to the standards WCAG Silver 3.0 and the Accessibility Conformance Testing Rules (ACT 1.0).

Crystal Scott is a Christian wife, mother of three extraordinary children, and a Certified Professional in Web Accessibility (CPWA). With a decade of frontend development experience and nearly six years specializing in accessibility, she has dedicated time to mastering Webflow and building accessible websites, with plans to launch Graceful Web Studio in 2025. She is also a design annotation ninja and the founder of the Accessibility Book Club, a thriving LinkedIn community. Crystal is pursuing a B.A. in graphic design and media arts at SNHU and is passionate about lifelong learning, technology, and user-centered design. She lives in Yakima, Washington, with her family and two blue nose pit bulls.

Chris McMeeking is an accessibility-focused engineer and the founder/CTO of MobA11y, where he leads teams in reimagining how AI can support people with disabilities. With over a decade of experience, Chris has shaped mobile accessibility at companies such as CVS Health and Deque Systems, building tools that help developers create more inclusive apps. His early work on Switch Control – now a core feature on iOS and Android – originated from a University of Michigan project, where he later served as a longtime mentor. Chris brings deep technical insight and human-centered leadership to every product he touches.

Dr. Keith Newton is a senior technology specialist and leader in digital accessibility, AI, and user-centered design. An invited expert in W3C Accessibility Platform Architecture and Maturity working groups, Dr. Keith has led enterprise-wide transformations by integrating AI and machine learning to improve accessibility testing, organizational maturity, and governance. He's collaborated with brands such as eBay, Warner Bros. Discovery, Salesforce, and American Express, developing scalable UX/UI solutions, building training programs, and promoting inclusive design. His deep expertise in WCAG compliance, assistive tech testing, and digital transformation has established him as a sought-after authority whose innovative contributions continue to shape the future of accessible technology.

Charlie Triplett is a UX engineering accessibility coach, championing inclusive UX design and accessible UI engineering. With two decades of experience in SEO and conversion optimization, he learned early on that the best digital products are the ones everyone can use. Now, he helps enterprises go beyond compliance, intertwining accessibility with innovation to create perceivable, operable, understandable, and robust experiences that drive both usability and business success. Charlie also wrote *The Book on Accessibility*, an operational guide designed to help organizations of any size integrate inclusion into their processes. Charlie lives in New York where he enjoys hiking, canyoneering, and a good bagel.

Kai Wong (she/they) is a passionate change agent for digital accessibility in the tech industry. They are an inventor (U.S. patent pending), a 2021 DC FemTech Award recipient, and have trained over 2,500 professionals. Dubbed a presenter who "could make stale bread interesting," Kai's strategy is simple: make it fun, and inspire action. They hold a B.S. in community health from the University of Maryland and certifications in accessibility, health education, and quality assurance.

About the reviewers

Daryl Suttie is a web accessibility professional with 25+ years of experience in HTML, CSS, and JavaScript, 10+ years in UX/UI and product management, and 10+ years of frontend testing. He has been focused on leveraging this experience specifically to support accessibility for 4+ years. He has worked in this capacity for major national retailer Canadian Tire, international consulting firm Accenture, and currently, the people platform company Dayforce (formerly Ceridian). He believes that education and mentorship are critical in shifting attitudes and practices in web accessibility. More importantly, he believes that "the work is the teacher," (i.e., the best learning is achieved through solving practical, relevant problems).

Ricky Osman is a principal technical writer for the Knowledge Center at TPG Interactive, a global digital accessibility company. In this capacity, he writes guidance for TPGi staff and clients conducting manual accessibility audits, as well as helping to craft the rules engine for TPGi's automated testing tools, developing training courses, and writing blog posts. Working remotely from his East Coast Australian home, he tests everything in his personal device lab to ensure he's not making things up. Ricky built his first website in 1994 when it was still assumed that accessibility was a given, and since then, has worked as a freelance web designer and frontend developer, and then as an accessibility specialist with organizations such as AccessIQ, Simply Accessible, AccessibilityOz, Intopia, Tenon, and now TPGi. Ricky has also spent his working life honing his writing skills, which has brought him work in senior editorial positions with the likes of UX Australia, SitePoint, Web Directions, and Smashing Magazine.

Table of Contents

Preface **xxiii**

Part 1: Foundations of Inclusive Design 1

Chapter 1: Defining Inclusive Design in the Digital Age 3

Understanding the historical context and evolution of digital accessibility 4

The shift from designing for disabilities to designing for human diversity • 5

The social model of disability • 6

Recognizing the diverse spectrum of user needs and abilities • 6

Spotting accessibility roadblocks in existing designs • 7

Exploring the seven principles of universal design 8

Implementing user-centered design methodologies 12

Understanding diverse user needs and experiences • 12

Techniques for inclusive user research and testing • 14

Iterative design processes • 15

Example: Microsoft's Xbox Adaptive Controller • 15

Recognizing the business case for inclusivity 17

Expanding market reach • 18

Driving innovation • 19

Enhancing brand reputation • 19

Legal compliance and risk mitigation • 20

Summary 21

Chapter 2: Understanding Diverse User Needs: Types of Disabilities and Assistive Technologies 23

Understanding disabilities: a spectrum of experiences **24**

Categorizing types of disabilities in digital contexts **25**

Visual disabilities: navigating without sight • 25

Auditory disabilities: amplifying access • 26

Motor disabilities: rethinking interaction • 27

Cognitive disabilities: simplifying complexity • 27

Examining the role of assistive technologies **28**

Types of assistive technologies and their functions • 28

Assistive technologies for visual disabilities • 29

Assistive technologies for auditory disabilities • 30

Assistive technologies for motor disabilities • 32

Assistive technologies for cognitive disabilities • 33

Integrating assistive technologies into design • 34

The impact of assistive technologies on design • 35

Mapping user needs to design solutions **35**

Understanding the user journey • 36

Real-world applications of mapping needs • 37

E-learning platforms: removing barriers to digital education • 37

E-commerce websites: ensuring inclusive online shopping • 38

Banking and financial tools: improving digital accessibility in finance • 39

Iteration: a key to success • 40

Identifying gaps and bridging them • 40

Centering lived experience: designing with, not for **41**

From observation to collaboration: building insight through lived experience • 41

Move from testing to partnership: integrating feedback into workflows • 42

From awareness to action: creating a culture of accountability • 43

Summary **43**

Chapter 3: Legal and Ethical Considerations in Accessible Design 45

Navigating global accessibility legislation 46

Establishing the legal foundation for digital accessibility • 47

Digital accessibility and global legal frameworks • 48

The challenges that remain • 50

Balancing legal compliance and ethical responsibility 51

The role and limitations of automated solutions • 51

Ethical dimensions of accessible design • 52

Moving beyond a tick-box exercise • 53

Forecasting future legal trends in digital accessibility 54

The increasing reliance on digital services • 54

Cultural shifts as a predictor for policy change • 55

The impact of an ageing population • 56

Addressing accessibility in AI governance • 57

Cultivating an ethically driven design culture 58

Awareness as the catalyst for change • 58

Education as a foundation for action • 59

Embedding accessibility into the design process • 59

Summary 60

References 61

Chapter 4: User Research and Testing for Inclusive Products 65

Recruiting diverse participants for inclusive research 68

Rethinking recruitment as a relational practice • 68

Beyond representation: designing from the edges • 69

Centering disabled lives through reciprocal recruitment • 71

Adapting research methods for accessibility 73

Creating safer and more empowered research spaces • 74

Embracing lived experience • 75

Inclusive research in practice, even when resources are tight • 76

Providing flexible communication methods • 77

Closing the loop with participants • 78

Research as a dynamic and human-centered practice • 79

Analyzing qualitative data from diverse user groups 80

Neurodivergence as method: how lived experience rewrites the rules • 81

The responsibility of meaning-making • 82

Are we interpreting in a way that reflects the weight of what was shared? • 82

Have we held the emotional, cultural, and systemic context, not just the quote? • 83

Have we left room for stories that don't align with our assumptions? • 84

Dynamic data in a fluid world • 84

Translating research insights into inclusive design solutions 86

Moving from fixes to futures • 87

Insights as relationships, not output • 88

Designing from disability, not for it • 89

Supporting designers with actionable findings • 90

Summary 91

Part 2: Designing for Accessibility 93

Chapter 5: Designing Accessible User Interfaces: Visual and Interaction Design 95

Using color within visual design 96

Using color for time-sensitive decision-making and safety • 96

Using color contrast to ensure legibility and clarity • 98

Using color in charts and diagrams • 99

Using color to convey errors • 100

Using color to denote action and interaction • 101

Designing with cognitive disabilities in mind 102

Consistency across the user interface • 102

Progressive disclosure and abstracting away unnecessary content • 104

Easy reversal of actions • 105

Designing for multiple alternative input methods 106

Voice commands to enable hands-free interaction • 107

Gesture controls to enhance navigation for touch-free interaction • 108

Assistive technologies, including support switches, eye tracking, and more • 109

Scaling designs for improved accessibility 109

Building a modular, accessible design system • 110

Elements that should be accessible at scale • 110

Empowering personalization through user customization • 111

Information architecture for accessibility • 112

Summary 114

Chapter 6: Creating Accessible Content: Writing, Images, and Multimedia 115

"The medium is the message" 116

Reviewing key actions for building an inclusive approach 117

Familiarize yourself with user needs • 117

Keep multiple user experiences in mind • 117

Learn from real users whenever you can • 118

Write all content in plain language • 118

Familiarize yourself with barriers and alternatives • 119

Making different types of content more accessible 119

Text • 120

Make pages easy to scan • 120

Plain language principles for text • 125

Images • 127

Types of images • 127

Utilizing alt text • 128

Avoiding text in images • 131

Format combinations for images • 132

Video and audio • 132

Closed captions • 132

Text transcripts • 133

Sign language • 134

Format combinations for video and audio • 134

Social media • 135

Quick case studies • 136

Case study 1: Sarah, a content author working with infographics • 136

Case study 2: Kavana, labelling buttons on a website • 137

Conducting the Needs Walkthrough exercise 138

Step 1: Gather a list of disability needs • 138

Step 2: Identify the standards and outcomes to test • 139

Walk through your content using WCAG • 139

Test against functional requirements • 140

Step 3: Plan your exercise event • 140

Step 4: Conduct the walkthrough exercise • 141

Action 1: Review the user stories and standards • 141

Action 2: Replicate each of the five user journeys • 141

Action 3: Record any barriers you faced • 142

Action 4: Record your solutions • 142

Step 5: Share what you've learned • 142

Step 6: Identify who solves the issues • 143

Summary 144

Chapter 7: Developing Accessible Websites: HTML, CSS, and ARIA Best Practices 145

Structuring content with semantic HTML 146

Defining non-semantic HTML • 147

Defining semantic HTML • 148

Why does semantic HTML matter? • 149

Styling for inclusive design with CSS 150

Ensure sufficient color contrast • 150

Avoid adding important content with CSS • 151

Respect user preferences • 152

Design for flexibility and scale • 153

Provide highly visible focus styles • 154

How CSS supports inclusive design • 156

Enhancing dynamics with ARIA 156

The five rules of ARIA • 157

aria-expanded • 158

aria-labelledby • 159

Applying ARIA to enhance dynamic and custom components • 160

Testing your code for accessibility conformance 160

Summary 163

Further reading 163

Chapter 8: Mobile Accessibility: Designing for Touch and Voice Interfaces 165

Mobile development accessibility considerations 166

Regarding iOS vs Android • 166

Regarding maintainability • 167

Optimizing touch targets for various abilities 169

The card pattern • 170

Two-dimensional scrolling • 171

Floating action buttons • 172

Extending the touchable area • 174

Utilize workflows • 174

Implementing alternative input methods 175

A simple design rule: platform controls or custom solutions • 176

Android keyboard support • 177

iOS keyboard support • 178

Understanding focus order in mobile accessibility • 178

Usability for all users • 180

Designing voice interfaces for accessibility 181

Identify controls • 183

Avoid nesting controls • 184

Ensuring responsive layouts across devices 185

Information density • 186

Adaptive layouts • 187

Utilize standard tools • 187

Summary 188

Part 3: Advanced Strategies and Implementation 189

Chapter 9: Accessibility in Emerging Technologies: VR, AR, and AI 191

Adapting VR experiences for various sensory abilities 192

Accommodations to improve VR accessibility • 193

Fictional case study: virtual escape room • 194

Enhancing AR accessibility through customization 196

Designing inclusive AR experiences • 196

Fictional case study: customizable AR overlays in a museum • 197

Leveraging AI to automate the inclusion of accessibility practices 199

Building inclusive AI: predictive text, algorithmic bias, and equitable design • 200

Predictive text • 200

Algorithmic bias in voice recognition • 200

Equitable AI design • 200

Fictional case study: AI-powered predictive text for users with motor disabilities • 200

Mitigating algorithmic bias in AI-driven interfaces 202

Data bias and representation • 203

Risk of ableism and overdependence on tool implementation • 203

Best practices for mitigating algorithmic bias • 204

Fictional case study using real-world dataset: FairFace – reducing racial bias in facial recognition for assistive applications • 206

Summary 207

Chapter 10: Foundations of Accessible Design System Patterns 209

Low vision **212**

Planning for dynamic typography • 213

Enabling web browser zoom • 213

Delivering intentional typographic patterns, not just assets • 213

Font scaling and font size: understanding the difference • 214

Text contrast • 214

Informative graphics contrast versus stroke width • 215

Guide color choices, don't just deliver a palette • 216

Affordances for magnification tools • 217

Color perception **218**

Deaf or hard of hearing **219**

Closed captions • 219

Transcripts • 220

Cognitive disabilities **220**

Use plain language in plain sight • 221

Affordances for text inputs • 221

Encourage autocomplete • 221

Discourage distractions • 221

Motor disabilities: designing for imprecise actions **222**

Motion sensitivity **223**

Respecting reduced motion settings • 223

Plan ahead for changes to motion • 223

Feature easy-to-find controls • 223

Photosensitivity **224**

Set clear guidelines • 224

Provide warnings and options • 224

Test for safety • 224

Additional tips • 224

Speech disabilities **225**

Motor disabilities: designing for assistive devices and software 226

Single point activation • 226

Accommodate landscape, portrait, and upside-down orientations • 226

Focus style definition • 227

Accommodate alternative navigation • 229

Very low vision or blindness 229

Choose a non-semantic typographic naming convention • 230

Provide annotation tools (but don't overdo it) • 231

Alternative text for informative versus decorative images • 232

Audio description for videos • 233

Intersectionality of disabilities 234

Features to omit for everyone with disabilities • 234

Tooltips • 235

Custom-styled select (aka listbox/combobox/dropdown) • 235

Carousels • 236

Summary 237

Chapter 11: Tools and Techniques for Accessibility Evaluation and Prioritization 239

Selecting appropriate automated testing tools 240

Performing effective manual accessibility audits 242

Keyboard navigation is key • 243

Get friendly with your screen reader • 244

Mobile device-specific screen reader tips • 245

Check descriptions of images • 246

Test the page in various states • 246

Zoom in on what matters: your product • 247

Color counts • 247

More essential checks • 247

Interpreting and prioritizing accessibility test results 249

Prioritizing bugs • 249

Blockers and serious issues • 249

Moderate issues • 250

Other factors to consider • 250

Dashboards • 251

Integrating accessibility testing into development workflows 252

Summary 255

Chapter 12: Building an Inclusive Design Culture: Strategies for Organizations 257

Cultivating empathy through immersive experiences 258

Understanding accessibility barriers firsthand • 258

Why disability simulations fall short • 260

Building empathy through real engagement • 261

Embedding empathy and driving company-wide commitment • 261

Turning empathy into action • 262

Integrating accessibility into workflows 263

Making accessibility a team responsibility • 263

Shifting left: building accessibility into the process from the start • 264

Reducing accessibility debt • 265

Measuring and reporting on accessibility progress 266

Defining success beyond compliance • 267

Choosing the right accessibility metrics • 268

Embedding accessibility into reporting • 269

Building transparency and accountability • 270

Sustaining progress over time • 270

Weaving continuous learning into inclusive design 271

Making accessibility training meaningful • 272

Encouraging peer learning and knowledge sharing • 272

Keeping up with evolving standards and tools • 273

Embedding accessibility into career growth • 273

Creating a culture of continuous learning • 274

Summary 274

Why subscribe? 277

Other Books You May Enjoy **279**

Index **283**

Preface

In an increasingly digital world, the ability to access information, services, and opportunities online is not a luxury – it's a human right. Yet, for millions of people with disabilities, everyday digital experiences still present barriers that prevent full participation. Whether it's a mobile app that can't be navigated with a screen reader, a website that hides key content behind inaccessible modals, or an AI tool that misinterprets a user's voice or intent, digital exclusion remains a persistent and solvable problem. *Inclusive Design for Accessibility* is a practical guide for anyone who wants to close that gap – designers, developers, content creators, QA testers, and organizational leaders alike.

This book brings together voices from across the accessibility community to provide you with a comprehensive roadmap for building inclusive digital experiences. It begins by grounding you in the principles of inclusive design and the diverse needs of real users. From there, it explores the nuts and bolts of implementation: how to design accessible user interfaces, write inclusive content, build accessible websites and mobile apps, and test with both automated tools and human insight. As technology evolves, so must our approach. That's why we also examine accessibility in emerging spaces such as VR, AR, and AI – and look at the organizational strategies needed to scale these efforts sustainably.

Throughout the book, you will encounter real-world tools and technologies – from screen readers such as VoiceOver and NVDA to development frameworks such as SwiftUI, Compose, and React Native. But more importantly, you'll gain a human-centered lens on accessibility: one that views inclusion not as a checklist, but as a creative, ethical, and business-critical practice. Whether you're just starting out or looking to level up your organization's accessibility maturity, this book will help you design with empathy, build with intention, and create digital products that work for everyone.

Who this book is for

This book is perfect for digital designers, developers, UX professionals, product managers, and business leaders committed to inclusive design. It offers practical skills to create accessible digital products, as well as covering legal and ethical aspects, user research, and building an accessibility culture within teams.

You don't need to be an expert to get value from this book. Each chapter offers actionable insights that stand on their own – be it UX design, product management, development, or accessibility leadership.

What this book covers

In *Chapter 1*, *Defining Inclusive Design in the Digital Age*, Dale Cruse charts the evolution of accessibility from a reactive add-on to a proactive, user-centered design mindset. The chapter introduces foundational concepts such as universal design principles, the social model of disability, and inclusive research methods, equipping you to design digital experiences that work for everyone.

In *Chapter 2*, *Understanding Diverse User Needs: Types of Disabilities and Assistive Technologies*, Dr. Angela Young, CPACC, examines how people with a wide range of permanent, temporary, and situational disabilities interact with digital technology. The chapter provides an in-depth look at assistive technologies, categorizes different disability types, and maps user needs to inclusive design strategies, offering you practical guidance for creating digital products that are truly accessible and empowering for all.

In *Chapter 3*, *Legal and Ethical Considerations in Accessible Design*, Maya Sellon explains how digital accessibility is both a legal requirement and an ethical imperative. The chapter outlines global laws such as the ADA and CRPD, explores the limitations of compliance-only approaches, and emphasizes the importance of designing inclusively from the start to foster meaningful and equitable user experiences.

In *Chapter 4*, *User Research and Testing for Inclusive Products*, Julianna Rowsell guides you through building inclusive research practices that center on disabled, neurodivergent, and marginalized participants from the outset. The chapter offers strategies for equitable recruitment, accessible methodologies, and meaningful analysis, emphasizing that inclusive research is a relational, ongoing practice – not a checklist – and must translate into design decisions that reflect the lived realities of diverse users.

In *Chapter 5*, *Designing Accessible User Interfaces: Visual and Interaction Design*, Nandita Gupta explores how to build interfaces that are inclusive across visual, cognitive, and input-related dimensions. The chapter provides actionable guidance on using color responsibly, supporting cognitive accessibility, enabling multiple input methods, and scaling accessible design systems, empowering you to create digital experiences that are not only usable but equitable for all.

In *Chapter 6*, *Creating Accessible Content: Writing, Images, and Multimedia*, Jennifer Chadwick empowers content creators to communicate inclusively by using plain language, accessible formatting, and alternative content formats. The chapter offers practical guidance for writing clear text, labeling interactive elements, describing images and multimedia, and concludes with a hands-on walk-through exercise to help creators evaluate and improve accessibility across content types.

In *Chapter 7*, *Developing Accessible Websites: HTML, CSS, and ARIA Best Practices*, Crystal Scott equips developers with foundational skills to build inclusive, accessible websites. The chapter covers semantic HTML, accessible CSS styling, proper use of ARIA attributes, and testing strategies – encouraging an accessibility-first mindset that integrates inclusion into every coding decision.

In *Chapter 8*, *Mobile Accessibility: Designing for Touch and Voice Interfaces*, Chris McMeeking explores the unique challenges and opportunities of building accessible mobile applications. The chapter offers practical guidance for optimizing touch targets, supporting diverse input methods, designing for voice interaction, and ensuring responsive layouts – empowering teams to deliver inclusive mobile experiences across platforms and devices.

In *Chapter 9*, *Accessibility in Emerging Technologies: VR, AR, and AI*, Dr. Keith Newton explores how to design inclusive experiences in immersive and intelligent systems. The chapter offers practical strategies for adapting virtual and augmented reality for users with diverse needs and highlights how AI can both enhance and complicate accessibility, depending on how it's implemented.

In *Chapter 10*, *Foundations of Accessible Design System Patterns*, Charlie Triplett lays out a comprehensive framework for building accessibility into design systems from the ground up. The chapter emphasizes starting with common user needs – such as low vision, color perception, and cognitive accessibility – before layering in more complex assistive technology support, guiding teams to create inclusive, scalable systems that reduce barriers by default.

In *Chapter 11*, *Tools and Techniques for Accessible Evaluation and Prioritization*, Kai Wong, CPACC, CHES®, guides you through the practical steps of accessibility testing for websites and mobile apps. The chapter covers how to combine automated tools with manual audits, prioritize and remediate issues, and embed accessibility into development workflows – helping teams build more inclusive digital products from the ground up.

In *Chapter 12*, *Building an Inclusive Design Culture: Strategies for Organization*, Denis Boudreau, CPWA, shows how to embed accessibility into an organization's daily operations, culture, and mindset. The chapter outlines strategies for cultivating empathy, integrating accessibility into workflows, measuring progress, and sustaining continuous learning – empowering teams to move from compliance to a meaningful, people-centered practice of inclusive design.

To get the most out of this book

This book is for people who've worked on digital products – designers, developers, content strategists, researchers, and beyond. We assume you understand the basics of building websites or apps, but not necessarily accessibility or inclusive design.

You don't need to be an expert. What matters is that you're ready to shift your perspective – to see inclusion not as a checklist, but as a core design value that benefits everyone.

If you bring curiosity, openness, and a willingness to question default practices, you'll find this book both practical and transformative.

Download the color images

We also provide a PDF file that has color images of the screenshots/diagrams used in this book. You can download it here: `https://packt.link/gbp/9781835888223`.

Conventions used

There are a number of text conventions used throughout this book.

`CodeInText`: Indicates code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and X/Twitter handles. For example: "Tags such as `<header>`, `<article>`, and `<section>` provide clear definitions of their role, allowing both browsers and assistive technologies to interpret and navigate the page effectively."

A block of code is set as follows:

```
@media (prefers-color-scheme: dark) {
  body {
    background-color: #121212;
    color: #f5f5f5;
  }
}
```

Bold: Indicates a new term, an important word, or words that you see on the screen. For instance, words in menus or dialog boxes appear in the text like this. For example: "**Semantic HTML** structures a web page using elements that inherently describe their purpose, making content more meaningful and accessible."

Warnings or important notes appear like this.

Tips and tricks appear like this.

Get in touch

Feedback from our readers is always welcome.

General feedback: Email `feedback@packtpub.com` and mention the book's title in the subject of your message. If you have questions about any aspect of this book, please email us at `questions@packtpub.com`.

Errata: Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you have found a mistake in this book, we would be grateful if you reported this to us. Please visit `http://www.packtpub.com/submit-errata`, click **Submit Errata**, and fill in the form.

Piracy: If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at `copyright@packtpub.com` with a link to the material.

If you are interested in becoming an author: If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, please visit `http://authors.packtpub.com/`.

Share Your Thoughts

Once you've read *Inclusive Design for Accessibility*, we'd love to hear your thoughts! Scan the QR code below to go straight to the Amazon review page for this book and share your feedback.

`https://packt.link/r/1-835-88823-2`

Your review is important to us and the tech community and will help us make sure we're delivering excellent quality content.

Download a Free PDF Copy of This Book

Thanks for purchasing this book!

Do you like to read on the go but are unable to carry your print books everywhere?

Is your eBook purchase not compatible with the device of your choice?

Don't worry, now with every Packt book you get a DRM-free PDF version of that book at no cost.

Read anywhere, any place, on any device. Search, copy, and paste code from your favorite technical books directly into your application.

The perks don't stop there, you can get exclusive access to discounts, newsletters, and great free content in your inbox daily.

Follow these simple steps to get the benefits:

1. Scan the QR code or visit the link below:

   `https://packt.link/free-ebook/9781835888223`

2. Submit your proof of purchase.
3. That's it! We'll send your free PDF and other benefits to your email directly.

Part 1

Foundations of Inclusive Design

This opening section lays the groundwork for designing accessible and inclusive digital experiences. Across four chapters, you will be introduced to the core principles, legal frameworks, and practical approaches that shape inclusive design in today's digital world. From understanding the evolution of accessibility and the spectrum of user needs to navigating global legislation and embedding ethical responsibility, these chapters reframe accessibility not as a checklist, but as a human-centered design commitment.

You will also learn how to conduct inclusive user research that respects lived experience, challenges extractive practices, and translates insights into actionable design decisions.

By the end of this part, you will be equipped with the knowledge, mindset, and foundational practices needed to begin building digital products that are not only compliant but truly inclusive.

This part includes the following chapters:

- *Chapter 1, Defining Inclusive Design in the Digital Age*
- *Chapter 2, Understanding Diverse User Needs: Types of Disabilities and Assistive Technologies*
- *Chapter 3, Legal and Ethical Considerations in Accessible Design*
- *Chapter 4, User Research and Testing for Inclusive Products*

1

Defining Inclusive Design in the Digital Age

By Dale Cruse

The journey of accessibility in design reveals a profoundly significant transformation in how we understand human diversity and technology's impact on society. We've moved beyond narrow, prescriptive approaches to embrace a more holistic view that celebrates the rich, complex spectrum of human abilities and experiences. This shift represents a design evolution; a fundamental reimagining of how technology can connect and empower people across different capabilities and contexts.

Understanding inclusive design is crucial in today's digital world: it's a professional necessity for creators, offers business advantages through expanded markets and innovation, ensures legal compliance, and helps build a more accessible digital society as technology becomes ever more central to daily life.

The overall approach is to provide a comprehensive foundation in inclusive design, shifting from viewing accessibility as an afterthought to seeing it as a fundamental design principle that benefits all users.

In this chapter, you'll explore the evolution of digital accessibility, the principles of universal design, and user-centered methodologies. You'll learn how to create inclusive products that empower users of all abilities, moving beyond accessibility as an afterthought to embedding it as a core design principle.

The goal is to provide a practical foundation for inclusive design, showing how it drives innovation and connection. You'll gain tools to rethink your approach, ensuring your work reflects the diverse needs of all users.

This knowledge is essential as technology becomes central to daily life. By embracing inclusive design, you'll not only enhance your professional skills but also contribute to a more equitable digital society where technology serves everyone.

So, in this chapter, we'll cover the following topics:

- Understanding the historical context and evolution of digital accessibility
- Exploring the seven principles of universal design
- Implementing user-centered design methodologies
- Recognizing the business case for inclusivity

Understanding the historical context and evolution of digital accessibility

In the past, accessibility was often a belated addition to design, with makeshift solutions tacked onto existing products. Frustrated users struggled with screen readers that couldn't interpret graphics, frequently mispronounced words, crashed often, and required manual updates for each new software version. Text-to-speech systems posed additional challenges with their robotic sound and poor comprehension. Rudimentary keyboard controls failed to work consistently across applications. Many magnification tools distorted text and images while significantly slowing down computers. Standalone screen readers were especially problematic – clunky software that struggled to play nice with mainstream applications. All these limitations delivered subpar experiences and reinforced the idea of disability as something "other" or apart from the norm. Fortunately, these solutions are a far cry from today's more seamlessly integrated systems.

As our grasp of disability and human diversity deepened, our design approach evolved too. The birth of the World Wide Web in 1989 brought fresh accessibility challenges and opportunities. A milestone came in 1999 with the publication of the Web Content Accessibility Guidelines, paving the way for standardized web accessibility practices. These guidelines introduced crucial standards like keyboard navigation, alt text for images, and proper heading structures – features that are now fundamental to web design. Their impact was revolutionary: for the first time, developers worldwide had a shared framework for creating accessible websites, leading to dramatically improved internet access for people with disabilities.

The shift from designing for disabilities to designing for human diversity

Before embracing a broader view of inclusive design, the field initially focused specifically on addressing disabilities through targeted solutions. Designers created specialized features like screen readers for blind users, closed captions for the deaf, and alternative input devices for those with motor impairments. These disability-specific adaptations were crucial first steps, helping many people access digital content and laying the groundwork for accessibility standards. However, this narrow approach often resulted in segregated solutions that marked disabled users as different, and the focus on specific impairments meant missing opportunities to create more universally beneficial features.

Designers realized that features meant for accessibility were helping everyone. Think about closed captions – they started out as a tool for folks who are deaf or hard of hearing, learning a new language, or navigating noisy environments, but wow, did they catch on! Now you've got language learners glued to them, picking up new words, and people in bustling cafes or gyms silently following their favorite shows. It's like they found a whole new audience.

Voice control features tell a similar story – originally designed for people with motor disabilities, they've become everyday conveniences for anyone cooking with messy hands, multitasking parents, or drivers needing hands-free operation. These once-specialized features are now so common that many don't even think of them as accessibility tools. This insight sparked a revolution in design thinking.

Inclusive design recognizes we're all on a sliding scale of abilities, changing with time and situations. Even someone with 20/20 vision might squint at tiny smartphone text under the blazing sun. Embracing these diverse scenarios allows us to create products that stand out for everyone.

This new mindset sees disability as fluid, not fixed. Historically, disability was often defined as a permanent, incapacitating condition – something that marked a person as fundamentally different from the norm. But this view has evolved. Today, we understand disability more broadly as a mismatch between an individual's abilities and their environment, shaped by permanent, temporary, or situational impairments. A broken arm, a parent juggling a baby, or a commuter on a bumpy bus – all face temporary hurdles that can limit interaction with digital systems. Inclusive design rises to these everyday challenges, making life smoother for everyone and reflecting a more nuanced understanding of human ability.

The social model of disability

At the heart of inclusive design lies a revolutionary concept: the social model of disability. Born from the experiences of disabled individuals, this model turns traditional thinking on its head.

Imagine a world where disability isn't a personal flaw, but a mismatch between people and their environment. That's the essence of the social model of disability. It stands in contrast to the earlier medical model, which defined disability as something "wrong" with an individual – a defect or deficiency to be fixed or cured. The social model flips this thinking, arguing that society's barriers – physical, attitudinal, and communicative – are the real disabling factors, not individual differences.

A wheelchair user isn't disabled by their wheels, but by stairs without ramps. A deaf person isn't limited by silence, but by a world that doesn't speak sign language. This perspective flips the script, placing the onus on society to adapt, not individuals to conform.

This isn't just some fancy idea on paper – it's a wake-up call! It's pushing us to weave accessibility into the very DNA of our world, from the ground up in our buildings to the nitty-gritty of our websites. No more slapping on accessibility features as an afterthought, like a bandage on a broken arm. We're talking about making it the real deal from day one.

In the digital landscape, this mindset is game-changing. When we encounter an inaccessible website, we now see a flawed design, not a "disabled" user. This shift sparks innovation, pushing us to create digital spaces that welcome everyone, regardless of ability.

Recognizing the diverse spectrum of user needs and abilities

A key skill in inclusive design is understanding the diverse spectrum of user needs and abilities, which includes:

- **Types of disabilities**: Recognizing that disabilities can be permanent, temporary, or situational, such as a permanent visual impairment or a temporary arm injury. For example, a situational disability might include a parent holding a baby while trying to navigate a website one-handed.
- **Accessibility spectrum**: Acknowledging that accessibility needs vary widely, necessitating flexible designs that accommodate different ability levels. For instance, some users may rely on screen readers, while others might need adjustable font sizes or color contrast options. This includes mental health conditions, which can affect concentration, memory, or stress levels – for example, limited-time interactions can induce anxiety or exclusion.

- **Intersectionality**: Considering how multiple factors can compound accessibility challenges. A deaf user who also has limited mobility, for example, may face unique barriers that require tailored solutions.
- **Changing needs**: Understanding that user needs evolve over time, especially as the population ages and faces age-related impairments. For instance, older adults may benefit from larger touch targets and simplified navigation to accommodate declining motor skills and vision.
- **Cultural diversity**: Take into account cultural and linguistic differences in user experiences. For example, designing for right-to-left language support or ensuring icons and imagery are culturally inclusive can make a product more accessible globally.

Honing this skill enables designers to create more inclusive products that attract a wider audience. This approach boosts market reach, drives innovation, enhances brand reputation, and reduces legal risks.

Spotting accessibility roadblocks in existing designs

Another key skill in inclusive design is the ability to uncover accessibility hurdles in current designs. These obstacles come in various flavors:

- **Visual**: Low-contrast text, color-dependent info, or images lacking alt text.
- **Auditory**: Videos without captions or transcripts, audio content without text backup.
- **Motor**: Tiny touch targets or keyboard navigation dead-ends.
- **Cognitive**: Jargon-heavy language, or headings that don't add meaning. Design choices like rapid-fire timed prompts or complex navigation can also increase cognitive load or trigger anxiety, particularly for users with mental health conditions.
- **Speech**: Voice-only interfaces without text alternatives, or speech recognition that struggles with diverse accents and speech patterns.

Spotting these roadblocks is like being part detective, part tech whiz, and part empath all rolled into one. You've got to get your hands dirty with assistive tech, keep your finger on the pulse of the latest accessibility guidelines, and – here's the kicker – bring real people with all sorts of abilities into your design playground. This goes beyond simply checking boxes; it involves deeply understanding how different people navigate the digital world.

While automated accessibility checkers can spot some issues, many barriers only surface through hands-on testing and user feedback. For instance, a bot might give a thumbs-up for alt text on all images, but it can't judge if that text paints a meaningful picture.

Designers and developers can craft inclusive digital experiences from day one, rather than playing catch-up with accessibility features down the road.

Inclusive design is about creating experiences that resonate across the full spectrum of human diversity. It is forward-looking and involves multiple abilities, contexts, and preferences that each person holds. This means bringing in the historical backdrop, embracing the social model of disability, and honing the skills for identifying and pulling down barriers in the path toward making sure the digital world works for everyone.

To put these principles into practice, let's explore one of the most influential frameworks in inclusive design: the Seven Principles of Universal Design, which provide concrete guidelines for creating truly accessible experiences.

Exploring the seven principles of universal design

In 1997, Ronald Mace and his team at North Carolina State University unveiled a game-changing framework: the *Seven Principles of Universal Design*. Originally crafted for physical spaces, these principles have found a new calling in our digital world, reshaping how we create inclusive online experiences.

Here are the seven principles:

1. **Equitable use**: This is the digital welcome mat. A website that lets everyone adjust text size and color contrast to their needs exemplifies equitable use in action.
2. **Flexibility in use**: Whether you tap, talk, or type, the truly flexible design accommodates the user. It's an app that feels just at home with your fingertips or voice commands.
3. **Simple and intuitive use**: A well-designed e-commerce site that guides you smoothly from browsing to checkout? That's simplicity and intuitiveness at work, making online shopping a breeze for everyone.
4. **Perceptible information**: A video player offering captions, audio descriptions, and transcripts? That's perceptible information in action, ensuring no one misses out on the content.
5. **Tolerance for error**: Think of a form that gently points out mistakes and shows you how to fix them. That's making sure a slip of the finger doesn't turn into a frustrating ordeal.

6. **Low physical effort**: A content management system with bulk actions and keyboard shortcuts embodies this principle, making digital tasks simpler.
7. **Size and space for approach and use**: This translates to designs that play nicely with all screens and fingers. Let's create touch targets that are easy to tap and layouts that work great whether you're on a phone or a desktop.

Embracing these principles goes beyond designing digital products; it allows us to create experiences that welcome everyone. That's when it dawns on us that features born from accessibility often turn out to be universal favorites, thus making our digital world better for all.

Bringing universal design principles to life in the digital world involves reframing our entire approach to digital experiences, ensuring that every interaction is intuitive, equitable, and empowering for people of all abilities. By prioritizing inclusivity from the start, we create digital environments that not only meet diverse needs but also foster innovation and connection for everyone.

The following examples illustrate how inclusive design principles can be applied in practice, showcasing how prioritizing accessibility and adaptability from the start leads to innovative, user-friendly solutions that benefit everyone.

- **Inclusivity from the get-go**: Think of diverse user needs as your design muse, not an afterthought. Apple's VoiceOver was built into iOS from its earliest versions, rather than being added later. Microsoft's Xbox Adaptive Controller was designed through continuous collaboration with gamers with disabilities, shaping its core features from the beginning.
- **Flex those layouts**: Create adaptable interfaces that look great on any device, from smartphones to desktop monitors. The BBC News website automatically adjusts its layout and image sizes across devices while maintaining readability. The Guardian's responsive design seamlessly shifts from a multi-column layout on desktop to a single column on mobile without losing content or functionality.

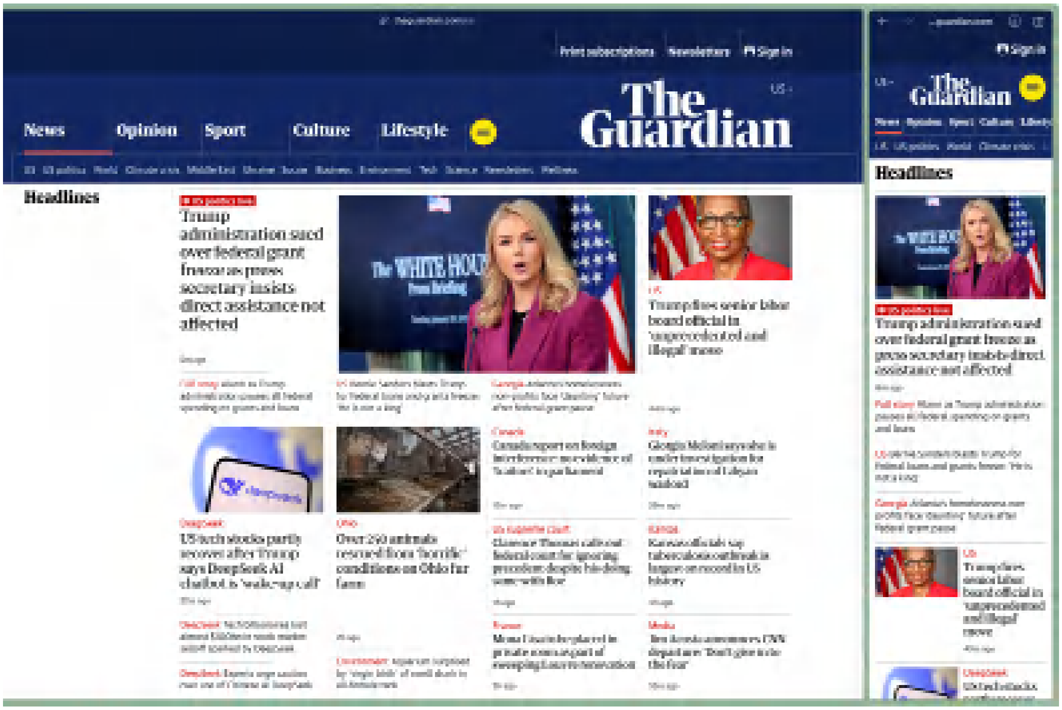


Figure 1.1: The Guardian's homepage on both the web and mobile

- **Multisensory magic**: Buffet-style interaction methods include visual, auditory, and tactile. Let the flavor be chosen by the user. Google Maps offers visual navigation, voice guidance, and haptic feedback for turns. Netflix provides audio descriptions, subtitles, and visual controls, letting users mix and match their preferred ways to consume content.
- **Keep it crystal clear**: Embrace simplicity in language and navigation. It's not dumbing down; it's opening. Gov.uk replaced complex government terminology with plain language, increasing user understanding. Duolingo uses simple, consistent navigation patterns and clear icons that help users focus on learning rather than figuring out the interface.

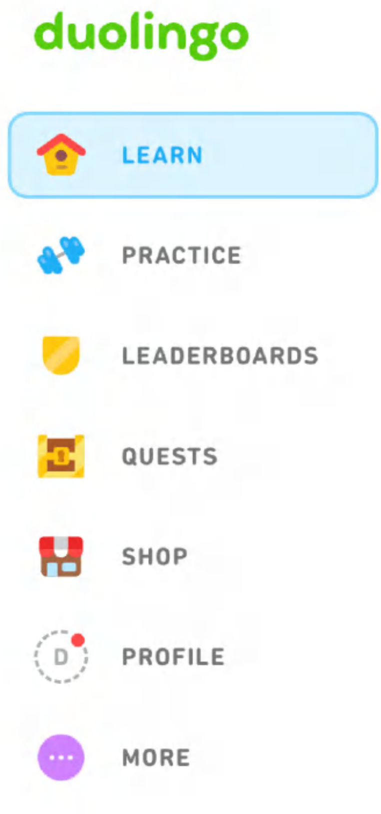


Figure 1.2: Duolingo's clear buttons and interface

- **Be a good player with assistive technologies**: Ensure your product plays along nicely with screen readers, voice recognition, and such. Twitter's image description feature works well with screen readers, making visual content accessible to blind users and those with low vision, though their needs can differ – for instance, users with low vision may rely more on zooming tools and contrast adjustments than fullscreen reading. Microsoft Word offers robust accessibility features, including properly labeled form fields and navigation aids that make documents more usable for screen reader users.

- **Diversity in testing**: Gather feedback from users across the ability spectrum. Their insights are design gold. Airbnb conducts usability testing with colorblind users to improve their map interfaces and property photos. Facebook's accessibility team includes disabled engineers who test new features before release.
- **Stay in the know**: Keep your finger on the pulse of accessibility guidelines. WCAG isn't just an acronym; it's your design compass. The adoption of these accessibility requirements led WordPress to update its editor for better keyboard navigation. Bank of America regularly updates its mobile app to meet evolving WCAG guidelines, ensuring continuous accessibility improvements.

These principles, woven into the fabric of digital design, create products that aren't just accessible but craft experiences that delight. It's all about recognizing that features born from accessibility often become universal favorites.

The Seven Principles of Universal Design aren't guidelines but a recipe for welcoming digital experiences. Embracing these principles throughout the design process means more than just checking off requirements; it opens doors to digital experiences that are more intuitive, flexible, and human for everyone, regardless of their abilities or life circumstances.

To bring these principles to life effectively, we need to understand how to implement user-centered design methodologies that put real people at the heart of the development process.

Implementing user-centered design methodologies

User-centered design puts people at the heart of the creation process, diving into the lives, needs, and experiences of all users and threading their perspectives through every thread in the fabric of design. By taking this approach, we're not just creating products, we craft experiences that resonate with human diversity. Accessibility isn't an afterthought, but the star of the show from day one.

Understanding diverse user needs and experiences

Understanding diverse users requires us to step beyond our own experiences and consider the many ways people interact with technology. That's the first leap in user-centered design – recognizing that our users are as diverse as a rainforest ecosystem, each with their unique abilities, preferences, and life contexts.

Think about it:

- One user might navigate the digital world with a permanent visual impairment.
- Another might be temporarily one-handed due to a sports injury.
- And yet another could be struggling with screen glare during their beach vacation.

To truly grasp this kaleidoscope of user needs, designers need to:

- **Become demographic detectives**: Dive deep into user research across a spectrum of backgrounds, ages, and tech-savviness levels. A designer might start by conducting structured interviews with users from different age groups and technical backgrounds, documenting how they navigate common tasks to identify patterns and pain points.
- **Master the ability alphabet**: From visual to auditory, motor to cognitive – consider the full range of human capabilities and challenges. For example, when designing a mobile app, a designer should test navigation both with standard touch controls and alternative input methods like voice commands or switch controls to ensure universal access.
- **Embrace the flux**: Remember, user needs aren't set in stone. They can shift with time, situation, or even mood. A designer can address this by creating flexible interfaces with adjustable settings, like a reading app that adapts its interface for different times of day or varying levels of user fatigue.
- **Connect the dots of intersectionality**: Recognize how different aspects of identity can intertwine, creating unique accessibility puzzles to solve. A designer might encounter a user who is both colorblind and uses keyboard navigation due to motor impairment, requiring careful consideration of both color contrast and keyboard focus indicators in the interface design.

Having recognized these diverse user needs and intersectional considerations, the next step is to develop effective research methods to understand them deeply. Let's explore specific techniques that help capture the full spectrum of user experiences and translate them into actionable design insights.

Techniques for inclusive user research and testing

To truly design for all, we need to dive deep into user experiences. Here are some powerful techniques to uncover those hidden insights:

- **Cast a wide net**: When recruiting participants, think about diversity in all its forms. A truly effective research panel should mirror the diversity of society itself, bringing together people of different abilities, ages, cultures, and tech-savviness levels, each contributing their unique perspectives.
- **Put yourself in their shoes**: The real contextual inquiry resembles being a fly on the wall in the users' lives. Designers do this by observing users in their natural environments, noting how they interact with technology in real-world contexts, and asking thoughtful questions to uncover insights.
- **Empathy bootcamp**: Empathy maps, personas, and other tools give your team permission to travel a mile in the users' shoes:
    - An **empathy map** is a visual tool that helps teams understand users by organizing observations into four quadrants: what they say, think, feel, and do. It captures users' behaviors, emotions, and motivations to build deeper empathy and inform design decisions.
    - A **persona** is a fictional, research-based representation of a target user, crafted to embody their goals, behaviors, needs, and challenges, helping teams align design decisions with the real experiences and priorities of their audience.
- **Accessibility detective work**: Regular audits of your digital products are like health check-ups. Use both high-tech tools and manual testing to sniff out any barriers lurking in your design.
- **Test driving tech**: Invite users who rely on assistive technologies to put your product through its paces. It's like having expert consultants who can spot issues you might never have considered.
- **Co-creation**: In effective design workshops, diverse users become more than just feedback providers – they're co-pilots in the creative process. These sessions can spark ideas that no designer alone could dream up.

Having gathered insights through co-creation with users, the next crucial step is to refine these ideas through systematic iteration. This ongoing process of prototyping, testing, and refining ensures that our collaborative insights evolve into truly effective solutions.

Iterative design processes

Design is a living, breathing entity that grows and adapts with each iteration. That's the essence of iterative design in user-centered methodologies. It's an act of creation, testing, and refinement in a circular process, with each iteration bringing us closer to the promise of inclusion.

This design process typically unfolds in four key moves:

- **Prototype playground**: We start by creating rough-and-ready prototypes, much like drafting a recipe. These early versions are designed to be tested and refined, inviting diverse users to provide their input, like having a group of chefs taste-test a dish still in development.
- **Feedback feast**: As the design evolves, we gather insights through user testing and analytics. This continuous feedback loop is like refining a recipe based on how people experience the flavors, textures, and presentation. Every interaction – clicks, swipes, and engagements – helps us adjust and improve.
- **Accessibility check-ups**: We regularly evaluate the design against accessibility standards, ensuring it works for everyone. Think of this as fine-tuning a recipe to accommodate different dietary needs, making sure the dish is enjoyable for all.
- **Embrace the plot twist**: Sometimes, user feedback reveals the need for significant changes. Just as a chef might rethink a dish entirely, we stay open to major revisions, pivoting when necessary to meet the needs of our diverse audience.

It's a journey of continuous improvement, where every cycle brings us closer to an accessible digital world.

Example: Microsoft's Xbox Adaptive Controller

The Xbox Adaptive Controller flips the script on what we might usually expect from design – it's a gaming controller that adapts to the user, not the other way around. It's not a controller, but a gateway to gaming for persons of limited mobility, engineered by Microsoft's Bryce Johnson, gamers with disabilities, and nonprofits.

Picture large, programmable buttons that connect to a universe of external switches, buttons, mounts, and joysticks. It's like a customizable command center for gaming, tailored to each user's unique needs.

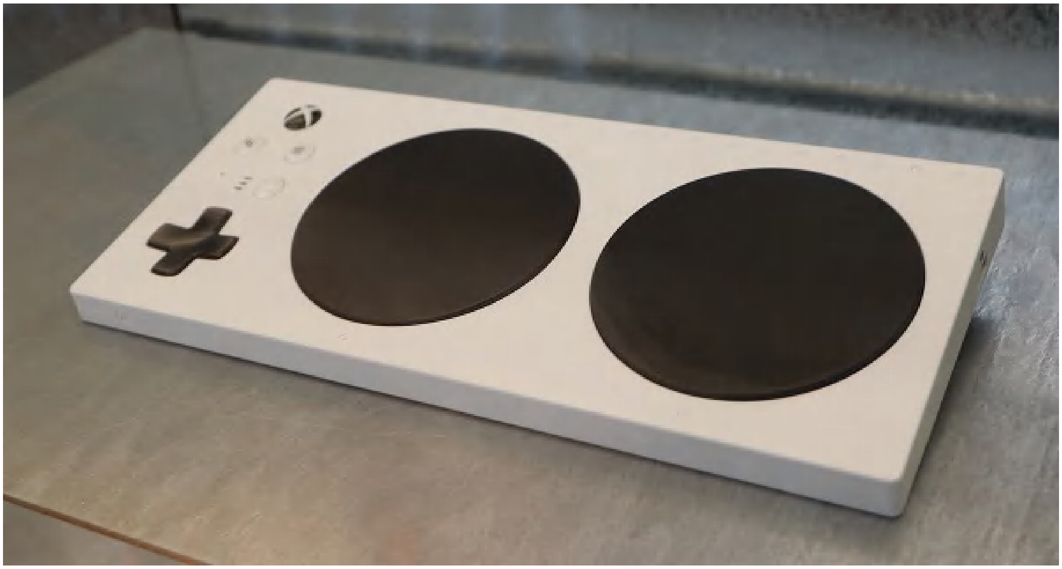


Figure 1.3: The Xbox Adaptive Controller is orbited by a host of assistive technologies to enable gameplay (Created by Geni, licensed under CC-BY-SA 4.0: `https://creativecommons.org/licenses/by-sa/4.0/`*)*

But what really sets this controller apart isn't just its cool features – it's the journey of its creation:

1. Microsoft dove deep into the world of gamers with disabilities, uncovering needs and challenges through in-depth research.
2. Occupational therapists and accessibility gurus became their co-pilots, guiding the design process every step of the way.
3. They crafted prototype after prototype, letting their target audience test-drive each version.
4. Feedback became their North Star, leading to major design pivots to boost usability and accessibility.
5. They went back to the big picture to make everything from packaging to setup as accessible as the controller itself.

The result? A game-changer that's not just serving its target audience – it's evening the playing field, proving that with proper design and insights from advocates, anyone can play.

While the ethical arguments for inclusive design are compelling, many organizations also need to understand its business impact to fully commit to implementation. Let's explore how inclusive design drives business success through expanded markets, increased innovation, enhanced brand reputation, and reduced legal risks.

Recognizing the business case for inclusivity

Inclusive design is the right thing to do – but did you know it's also a smart business move? It directly improves business performance by creating digital products that work for everyone. According to a 2016 report from Forrester and Microsoft, accessible design can improve customer satisfaction, enhance brand perception, and increase market reach. Plus, companies that invest in accessibility are more likely to foster innovation and employee productivity.

When we design with diverse users in mind, we expand our market reach, increase user satisfaction, and build more innovative solutions:

- **Bigger audience, bigger profits**: You roll the red carpet out to more customers when you design for all. By addressing the needs of underserved communities, you tap into new markets and unlock significant revenue potential.
- **Innovation ignition**: Designing inclusively challenges your team to think outside the box, often leading to fresh solutions. This creative problem-solving can spark breakthroughs that benefit your entire product line and set you apart from competitors.
- **Brand boost**: Show the world you care and watch your reputation soar. It's like a PR campaign that actually means something. Consumers are increasingly drawn to brands that align with their values, and inclusivity can become a powerful differentiator.
- **Legal safety net**: Stay ahead of accessibility regulations and keep those lawyers at bay. Proactively embracing accessibility not only minimizes legal risks but also positions your company as a responsible industry leader.
- **Universal appeal**: Here's the kicker: features designed for accessibility often end up delighting all users. From captions to voice commands, these innovations enhance usability for everyone, creating a more intuitive and enjoyable experience.

Making a strong business case for inclusivity elevates design from optional to essential. Showing how accessible design opens new markets and improves products grabs decision-makers' attention. Here are a couple of examples:

- Audiobooks that provide access for visually impaired users. Audiobooks have become a booming industry, proving that accessibility features can drive revenue while empowering users. They also benefit multitaskers, commuters, and those with learning differences, showcasing how inclusive design can expand a product's reach and utility.
- Closed captions that help everyone in noisy environments. Originally designed for the deaf and hard-of-hearing community, captions are now widely used in gyms, airports, and on social media, enhancing the user experience for all. This example highlights how accessibility features often create universal benefits, increasing engagement and customer satisfaction.

Inclusive design merges ethics with smart business. Addressing diverse needs creates better products and fosters brand loyalty.

Expanding market reach

Inclusive design unlocks a powerful market that's been hiding in plain sight. That's what inclusive design does – it's like finding a secret passage to a whole new world of customers. Let's crunch some numbers that'll make your brain light up:

- **A billion-strong audience**: The World Health Organization drops a bombshell – over 1 billion people worldwide have some form of disability. That's 15% of our global family!
- **Disposable income goldmine**: In the U.S. alone, people with disabilities are sitting on a $490 billion treasure chest of disposable income, according to a 2018 report by the American Institutes for Research (`https://www.air.org/sites/default/files/2022-03/Hidden-Market-Spending-Power-of-People-with-Disabilities-April-2018.pdf`). That's not pocket change – it's a market waiting to be served.
- **Tech market on fire**: The global assistive technology market is set to explode, reaching a whopping $88 billion by 2034, according to Market Research Future (`https://www.marketresearchfuture.com/reports/assistive-technology-market-29777`). Talk about a growth opportunity! Those accessibility features you're considering? They're not just for a niche audience. They're the secret sauce that could make your product irresistible to a much wider crowd. A ramp serves far more than wheelchairs – it becomes invaluable for strollers, luggage, and delivery carts alike.

Embracing inclusive design means making a smart business choice while also doing good. You're opening doors to markets you didn't even know existed and creating products that could become the new gold standard for usability.

Driving innovation

Inclusive design sparks creativity that elevates products for everyone. When designers solve challenges for diverse users, they often uncover solutions that benefit the entire user base.

Just as wheelchair ramps help everyone from parents with strollers to delivery workers, digital accessibility features create widespread improvements:

- **Voice assistants**: From accessibility tools to universal convenience. People now regularly use voice commands to set timers while cooking, make calls while driving, or control smart home devices from across the room – tasks that were originally designed to help those with limited mobility.
- **Predictive text**: Originally supporting motor-impaired users, now a time-saver for all. This feature has evolved from helping users with motor difficulties type more efficiently to becoming an essential productivity tool that speeds up messaging and reduces typing errors for millions of users daily.
- **High-contrast mode**: Crucial for visual impairments, helpful in bright sunlight. This accessibility feature not only makes screens readable for users with visual impairments but has become invaluable for anyone trying to read their phone outdoors or reduce eye strain during prolonged screen use.

Designing this way unlocks innovations that enhance experiences for everyone. It focuses on creating better solutions for many, rather than just accommodating a few.

Enhancing brand reputation

In today's market, where social responsibility is a growing expectation, a commitment to inclusivity sets brands apart. This approach can foster increased loyalty and generate positive word-of-mouth marketing. Companies that prioritize inclusive design, like Nike, often enjoy a boost in positive press and consumer goodwill. By aligning with values of equity and accessibility, brands not only strengthen their reputation but also position themselves as leaders in their industry, earning trust and admiration from a broader audience.

Legal compliance and risk mitigation

Digital accessibility isn't just a best practice – it's a legal requirement in many regions. Laws like the Americans with Disabilities Act in the U.S. and the European Accessibility Act mandate that companies ensure their digital products are accessible to all users. Embedding inclusive design from the start allows businesses to avoid costly lawsuits and the headaches of retrofitting. This proactive approach also positions them to easily adapt to future accessibility regulations. Embracing inclusivity isn't just smart; it's a proactive strategy for compliance and risk management.

Figure 1.4. San Francisco street art celebrating the Americans with Disabilities Act. Mural by Dan Fontes. Photo by Dale Cruse.

Summary

This chapter has explored the fundamental principles of inclusive design, tracing its evolution from basic accessibility features to a comprehensive approach that benefits all users. You've learned about the Seven Principles of Universal Design and how they apply to digital products, discovered techniques for implementing user-centered design methodologies, and understood how to build a compelling business case for inclusivity. As the chapter revealed, *inclusive design isn't just about making technology accessible – it's about making technology better for everyone*.

This knowledge is crucial in today's digital landscape, where digital accessibility increasingly determines participation in society. Understanding inclusive design principles helps create better products for everyone while expanding market reach and reducing legal risks. Moreover, these skills are becoming essential for designers, developers, and product managers as organizations recognize that inclusive design drives innovation and enhances user experience for all.

This chapter lays the foundations for more detailed explorations. As we move forward, we'll dig deeper into the unique needs of different user groups, including a closer look at specific disabilities, mental health considerations, and the assistive technologies that support inclusive access.

2

Understanding Diverse User Needs: Types of Disabilities and Assistive Technologies

By Dr. Angela Young, CPACC

If you opened a website today and couldn't read the text, hear the audio, or click the links, how would you find the information you needed? For millions of people like myself, that's not hypothetical; it's an everyday reality. Accessibility bridges that gap. It turns barriers into opportunities and ensures that digital spaces are open and welcoming to everyone.

Accessibility is more than standards and checklists; accessibility is about people. It's about understanding how different needs shape the way we interact with the world. Maybe you've relied on captions to follow a video in a noisy café or adjusted font size on your phone when you forgot your glasses. These simple, everyday tools reflect the same principles that make inclusive design so powerful. Compliance ensures a baseline, but true accessibility means designing with real users in mind.

In this chapter, we'll break down the diverse ways people experience the digital world, the role of assistive technologies, and how thoughtful design can remove barriers. You'll learn how different types of disabilities affect digital interactions, how inclusive design benefits everyone, and how small changes can make a significant impact. Whether you're a designer, developer, or decision-maker, understanding these concepts will help you create digital experiences that are not just functional but truly inclusive. Accessibility isn't an afterthought. It's a fundamental part of building technology that works for all.

In this chapter, we will cover the following topics:

- Understanding disabilities: a spectrum of experience
- Categorizing types of disabilities in digital contexts
- Examining the role of assistive technologies
- Mapping user needs to design solutions
- Centering lived experience: designing with, not for

Understanding disabilities: a spectrum of experiences

Like so many other identities within the human experience, disability exists on a spectrum:

- Situational disabilities are tied to context, such as trying to read your phone in glaring sunlight or holding a baby while navigating a tablet with one hand
- Temporary disabilities result from short-term conditions, such as the recovery period from an injury or surgery
- Chronic conditions, such as arthritis or migraines, fluctuate, creating good days and bad
- Permanent disabilities, such as blindness or paralysis, are ongoing but just as varied in how they shape people's interactions

Beyond definitions, it's important to recognize how fluid and often invisible these experiences can be. A person may move between different categories over the course of their life… or even throughout a single day! Pain flare-ups, fatigue, changes in mobility, or sensory overwhelm can all shift how a person navigates the digital world from moment to moment.

For example, a student with fluctuating vision due to a chronic condition might need high-contrast settings and enlarged text to complete their homework on bad days but have no issue on good ones. If those options weren't available, how would they stay engaged with their studies?

Or, consider a new parent who temporarily loses hearing clarity due to sleep deprivation and environmental noise. At that moment, real-time captions on a video call might be an essential feature. These small accommodations ripple outward and benefit a wide audience.

When we understand that everyone faces barriers at some point in their lives, accessibility becomes universal – wheelchair ramps aid delivery workers and parents with strollers, while closed captions support diverse needs from noisy environments to language learners.

This is the foundational idea behind inclusive design: that accessibility is not an exception, but an expectation. By embedding flexibility and choice into our digital environments, we empower all users – not only those who are disabled by medical definition.

Importantly, disability is not solely medical; it's also social. It arises from the mismatch between someone's body or mind and the environment or systems that they navigate. A person may not be disabled by their condition alone, but by a website that does not support keyboard navigation, or an app that relies solely on sound without alternative text. That perspective shifts the responsibility: from asking individuals to adapt to inaccessible systems to asking systems to meet the needs of real people.

As we move forward, it's helpful to group disabilities into broader categories to understand how they impact digital interaction. After all, instead of framing it as addressing the needs of a few, we open up digital opportunities when we design for all. The next section explores these categories – visual, auditory, motor, and cognitive – and introduces design strategies that support each one.

Categorizing types of disabilities in digital contexts

When designing digital products, it's crucial to understand the diverse ways people interact with them. Disabilities are not isolated conditions – they are part of the human experience and influence how individuals perceive, process, and engage with technology. Due to this, it is important to note that "disabled" is an identity group that can intersect with any other identity. You can be Muslim and disabled, queer and disabled, Gen Z and disabled, and so on. The list goes on and on.

Under the disability umbrella, the four main categories of disabilities – visual, auditory, motor, and cognitive – encompass a wide range of conditions and present unique challenges. Addressing these challenges requires thoughtful design that considers the full spectrum of human ability.

Visual disabilities: navigating without sight

For users who are blind or have low vision, navigating a digital interface involves translating visual information into other forms, such as audio or tactile feedback. But there can be problems. A screen reader user, for example, depends on properly structured code to navigate a web page. If the site lacks semantic headings or descriptive alt text for images, it can feel like you're at the carnival navigating through a mirror maze.

Imagine a job application portal where fields aren't labeled properly. A user – perhaps someone who is blind – relying on a screen reader might hear "Edit Box" repeatedly without context, making it impossible to complete the form. Contrast that with an accessible form where each field has a clear label, such as "First Name," "Email Address," or "Upload Resume." The difference, then, transcends functionality through empowerment.

Low-vision users who prefer alternative methods such as zoom, rather than screen readers, may face different but equally significant barriers. For example, text that runs off-screen when zoomed can make text unreadable, while fixed font sizes may render content inaccessible on smaller screens. Responsive design and scalable text options are straightforward solutions that significantly enhance usability.

Auditory disabilities: amplifying access

Auditory disabilities include a wide range of experiences – deafness, being hard of hearing, and auditory processing disorders. For example, as someone who is deaf, I can tell you that when videos lack captions or podcasts don't include transcripts, we're not just inconvenienced – we're shut out entirely.

Accessibility efforts must also account for those who are hard of hearing, and who may rely on volume controls, amplification tools, or hearing aids. These users need content that avoids audio distortion and supports user-controlled sound settings. Others may experience difficulty processing sound accurately, even if hearing tests place them within "normal" ranges.

Designing for auditory accessibility broadens usability for many people beyond those with documented disabilities. Consider an international student trying to follow a fast-paced lecture in their second language, or someone watching a video tutorial in a crowded café. Captions, transcripts, and visual alternatives to sound benefit all of these users while serving as critical access points for those of us in the d/Deaf community.

Auditory inclusion extends beyond media content. Systems that use visual alerts – such as flashing lights or on-screen notifications in place of alarms – help ensure everyone receives the same information, regardless of hearing ability.

> It is important to note that one experiences deafness but identifies as part of the Deaf community. The condition is lowercase and the identity group itself is capitalized in writing.

Motor disabilities: rethinking interaction

Motor disabilities encompass a broad range of conditions, from tremors and arthritis to paralysis and amputation. These conditions affect how users physically interact with devices, whether through a mouse, keyboard, or touchscreen.

Consider a user with limited dexterity who navigates using a head pointer or switch controls. Every click or movement requires effort, so overly complex workflows or small, difficult-to-click buttons create unnecessary barriers. Designing with simplicity and efficiency in mind (such as providing large touch targets and minimizing the number of interactions needed to complete a task) makes technology more accessible.

Voice control is another transformative tool for users with motor disabilities. However, voice-enabled interfaces must account for variability in speech patterns. For example, a person with a speech impediment or even someone with a non-native accent may have difficulty being understood by certain systems. Providing multiple input options ensures no one is left out.

Cognitive disabilities: simplifying complexity

Cognitive disabilities – such as ADHD, dyslexia, and intellectual disabilities – affect how users perceive, process, and act on information. A cluttered web page with inconsistent navigation can easily overwhelm someone with ADHD, while long blocks of dense text may discourage users with dyslexia from engaging altogether.

Simplifying complexity makes a difference. Breaking content into smaller, digestible chunks, using consistent design patterns, and incorporating visual aids can help users stay focused and process information more effectively. For example, a health insurance website that uses plain language and intuitive icons to explain coverage options is far more accessible than one buried in industry jargon.

Interactive content can also support cognitive accessibility. A budgeting app that uses color-coded graphs to show spending habits provides visual clarity for users who may struggle with numerical data alone.

It's equally important to consider psychiatric and mental health disabilities, which are often mistakenly grouped with cognitive impairments but differ in how they impact digital interaction. Anxiety, PTSD, depression, and similar conditions can make certain design choices – like timed forms, pop-ups, or unexpected motion – deeply distressing or outright unusable. For instance, a user with an anxiety disorder may abandon a form altogether if it includes a countdown timer, even if they know the task is simple.

Recent updates in WCAG 2.2 have begun to address these issues more explicitly. Success criteria such as 3.3.7 Redundant Entry, which avoids making users re-enter information, and 3.2.6 Consistent Help, which ensures guidance is readily available, were developed with mental health-related needs in mind. These changes reflect a growing awareness that accessibility isn't just about comprehension – it's also about emotional safety and reducing cognitive load.

Designing for these needs often leads to better experiences for everyone. High-contrast modes help people use devices outdoors. Captions are useful in noisy environments. Voice control supports users who are cooking or multitasking. These aren't edge-case fixes. They're universal enhancements.

True accessibility goes beyond recognizing barriers – it means actively removing them. Assistive technologies, from screen readers and magnifiers to eye-tracking and switch controls, make digital interactions possible for millions. Understanding how they work – and designing with them in mind – is key to building truly inclusive, human-centered experiences.

Examining the role of assistive technologies

We've discussed how you might navigate a website if you couldn't see the screen or send a text if you couldn't use your hands. Now, let's go a little further. Assistive technologies answer these questions by transforming barriers into pathways. These tools empower individuals to interact with technology in ways that meet their specific needs, making the digital world more inclusive and accessible.

Types of assistive technologies and their functions

Assistive technologies – tools, devices, or software designed to support individuals with disabilities in accessing, interacting with, and navigating digital and physical environments – aren't "extras" or "nice-to-haves"; they are essential. These technologies not only provide access but also shape the design decisions we make when we keep them in mind throughout the design process. When we design with a desire for our digital environment to have streamlined use capabilities with assistive technologies, we ensure that digital spaces are usable for everyone.

Understanding how these technologies work helps designers and developers create digital experiences that are not only accessible but also seamless and intuitive for all users.

Assistive technologies for visual disabilities

For individuals with visual disabilities, accessing digital content requires alternative ways to perceive and interact with information. Assistive technologies in this category transform text, images, and interfaces into accessible formats that support independent navigation and engagement.

Screen readers: making the digital audible

Screen readers are assistive technologies that convert on-screen text into spoken words, allowing users who are blind or have low vision to navigate websites, apps, and digital documents. Tools such as JAWS, NVDA, and VoiceOver interpret the structure of a web page and communicate it audibly to the user.

Some users – particularly those who are DeafBlind – pair screen readers with **refreshable Braille displays** (**RBDs**), which translate screen reader output into tactile Braille. While screen readers handle the processing of content, RBDs provide an alternative output format that makes digital environments accessible through touch.

To function effectively, screen readers benefit significantly from semantic HTML and well-implemented ARIA. Clear headings, landmarks, labels, and roles help these tools convey meaningful structure and context to the user. Without this structure, the user experience can become confusing or incomplete.

For example, a screen reader can identify headings, links, and form fields when they're coded correctly. However, when a button has no accessible name at all, a screen reader may skip over it entirely or announce it generically – such as "button," with no context – leaving users to guess what it does.

Even when buttons are labeled, the label itself still matters. A button labeled "Button123" may technically meet the requirement for a name, but it tells the user nothing. Imagine trying to shop online and hearing "Button123" instead of "Add to Cart." Would you click it, not knowing whether it adds an item to your cart – or even charges your credit card?

Accessible design isn't just about making something navigable – it's about making it clear. A well-structured page with meaningful, descriptive labels turns a confusing experience into an intuitive one.

Screen magnifiers and high-contrast tools

For users with low vision, **screen magnifiers** enlarge on-screen content, while high-contrast tools enhance visibility by increasing the contrast between text and background colors. These features are critical for ensuring readability and usability.

Many of us know how difficult it is to read a screen after having our pupils dilated at an eye exam. Imagine trying to complete an urgent banking transaction while struggling to read tiny text that doesn't scale properly. A financial dashboard must remain functional when zoomed via the browser to 200%. If buttons overlap or text becomes unreadable at high magnification, the user is effectively excluded. Responsive design practices ensure that content adapts gracefully, regardless of the settings.

Braille displays: tactile access to digital content

RBDs convert digital text into Braille characters, enabling blind users to read content through touch. These devices are especially important for users who are DeafBlind or for those who prefer tactile reading over audio output from screen readers. Braille displays work in tandem with screen readers to translate content into a tactile format, offering precise, silent access to digital information.

To ensure compatibility, websites and apps must use clean, semantic HTML and properly implemented ARIA, allowing screen readers to pass meaningful information to Braille devices. Without this structure, content may be skipped, misrepresented, or rendered unreadable.

Audio description for low-vision users

For multimedia content, audio descriptions serve a parallel purpose. **Audio descriptions** narrate key visual elements – such as scene changes, body language, or text on the screen – that are not captured in standard dialogue. This provides blind users with critical context, especially in educational or storytelling environments.

Together, tactile access and audio description expand access for blind users across a wide range of digital experiences – not just reading but watching, navigating, and interacting.

Assistive technologies for auditory disabilities

People with auditory disabilities rely on visual and text-based alternatives to access spoken content, alerts, and communication tools. Assistive technologies in this category bridge the gap by ensuring that information typically conveyed through sound is accessible in other formats.

Captions and transcripts: ensuring audio accessibility

For users who are deaf or hard of hearing, captions and transcripts provide essential access to multimedia content. **Captions** display spoken words and relevant sound cues on videos, while **transcripts** provide a full text version of audio-based content such as podcasts and recorded meetings.

However, not all captions are created equal. Automatically generated captions often contain errors, leading accessibility advocate Meryl Evans to refer to them as "craptions." Providing accurate, human-reviewed captions ensures that deaf and hard-of-hearing users can fully engage with content.

Visual alerts: replacing sound-based notifications

Instead of relying solely on auditory cues, **visual alerts** – such as flashing lights or on-screen text notifications – can provide alternative ways for users to receive important information. For example, a system alert that typically relies on a sound notification should also display a visible message to ensure accessibility.

Hearing aids and cochlear implant compatibility

Many digital platforms integrate with assistive listening devices such as Bluetooth-enabled hearing aids and cochlear implants. Compatibility isn't just about the device itself – it's about how the audio content is delivered.

For users relying on hearing aids or cochlear implants, high-quality audio files, minimal background noise, and clear, evenly paced speech are essential. Using standard, uncompressed, or lightly compressed formats (such as MP3, AAC, or WAV) helps ensure that assistive devices can process and transmit sound clearly. Audio that is overly compressed or distorted may not work well with hearing aid processors, which are often optimized for speech frequencies.

In environments where **Telecoil (T-coil) systems** are supported – such as theaters, classrooms, or kiosks – ensuring that your hardware outputs sound via inductive loop systems or is compatible with assistive listening transmitters can make all the difference. While T-coil support is largely hardware-based, digital systems that interface with the hardware (such as public address systems or kiosks) must output clean audio signals through supported channels.

In short, accessible audio is clear, properly formatted, and delivered in a way that integrates smoothly with assistive tech – whether through Bluetooth, loop systems, or direct audio input.

You can learn more about hearing loss and assistive tech for deaf/hard-of-hearing users through the following resources:

- *Hearing Loss Association of America (HLAA) – Assistive Listening Systems Overview:* `https://www.hearingloss.org/find-help/hearing-assistive-technology/assistive-listening-systems/`
- *Apple's Made for iPhone Hearing Aids Program:* `https://support.apple.com/en-us/108780`
- *National Association of the Deaf – Technology Recommendations:* `https://www.nad.org/resources/technology/`

Assistive technologies for motor disabilities

For individuals with motor disabilities, interacting with digital devices may require alternative input methods beyond a traditional keyboard and mouse. Assistive technologies in this category provide solutions that enable users to navigate, type, and control digital interfaces with ease.

Voice recognition software: hands-free interaction

For users with motor impairments, **voice recognition** software offers a way to interact with devices without using a keyboard or mouse. Tools such as Dragon NaturallySpeaking allow users to dictate text, navigate menus, and execute commands using spoken instructions.

However, this functionality depends on interfaces that are voice-friendly. Imagine a user dictating "Click Submit" but encountering a button that isn't labeled with a recognizable term. By using clear, predictable labels and simplifying workflows, designers ensure these tools work seamlessly.

Adaptive input devices: expanding interaction options

Adaptive technologies such as **switch controls**, **sip-and-puff systems**, and **eye-tracking software** redefine how users interact with digital devices. These tools are vital for individuals with severe mobility impairments who cannot use traditional input methods.

For instance, a person using a single switch control might navigate an interface by cycling through options and selecting one with a click. If the interface is overly complex or requires too many steps to complete a task, the user's experience becomes frustrating. This can be especially difficult on mobile devices, where someone with a fine motor disability would struggle to select small buttons. Large touch targets, streamlined navigation, and clear workflows ensure that adaptive devices can be used effectively.

Keyboard navigation and alternative keyboards

Many users with motor disabilities rely on **keyboard navigation** rather than a mouse. Designing interfaces that allow users to fully navigate using only a keyboard, such as ensuring logical tab order and visible focus indicators, ensures accessibility. Some individuals may also use alternative keyboards, such as one-handed keyboards or on-screen keyboards, to interact with digital content.

Assistive technologies for cognitive disabilities

Cognitive disabilities affect how individuals process, understand, and interact with digital content. Assistive technologies in this category focus on enhancing comprehension, reducing cognitive load, and improving usability for a diverse range of users.

Text-to-speech tools: supporting comprehension

Text-to-speech (TTS) tools convert written text into spoken words, benefiting users with cognitive disabilities, learning disabilities (such as dyslexia), or low literacy. These tools help users process and retain information more effectively.

For example, a student with dyslexia may struggle with dense academic text but can follow along more easily when using a TTS tool that reads the content aloud. However, if a website contains poorly structured content or inconsistent formatting, TTS tools may struggle to convey information clearly.

Speech-to-text tools: reducing cognitive load

Speech-to-text (STT) tools transcribe spoken words into text, providing an alternative communication method for users who may struggle with written language or typing. These tools are essential for individuals with conditions such as dysgraphia or auditory processing disorders.

Ensuring that digital platforms support STT functionality and provide clear error correction options enhances accessibility for users who rely on these tools.

Simplified interfaces and readability enhancements

Some users experience cognitive or information-processing difficulties, including people with ADHD, intellectual disabilities, or learning differences such as dyslexia. Others, such as autistic users, may benefit from predictable design patterns and reduced sensory overload, even though autism is not itself a cognitive disability but can co-occur with cognitive disabilities.

Designing interfaces that minimize cognitive load can significantly improve accessibility. Helpful features include the following:

- Consistent navigation to reduce confusion
- Plain language and clear instructions to support understanding
- Content chunking to break text into manageable sections
- Icons and visual aids to reinforce meaning

For example, an online tax preparation tool that presents users with long, scroll-heavy pages of dense financial terminology can quickly become overwhelming – especially for those with ADHD, dyslexia, or processing disorders. By breaking the process into step-by-step screens with progress indicators, plain language explanations, and visual cues (such as icons or tooltips), the tool becomes far easier to navigate and complete with confidence.

This kind of inclusive interface design doesn't rely on assistive technologies, but it does share the same core goal: making content easier to access, understand, and use. As we turn to the next section, we'll focus on the tools that actively assist users in navigating digital environments – and how to design with those technologies in mind from the start.

Understanding the range of assistive technologies is just the first step. Ensuring they function seamlessly within digital environments is where inclusive design truly takes shape. These tools can only be as effective as the systems they interact with, which is why designing with accessibility in mind from the start is essential. In the next section, we'll explore how to integrate assistive technologies into the design process, ensuring digital experiences are not just technically accessible but truly usable for everyone.

Integrating assistive technologies into design

Assistive technologies are only as effective as the environments in which they operate. A screen reader can navigate a site better if it has been coded with accessibility in mind, while voice recognition software works best with interfaces designed to accommodate verbal commands.

Of course, we can't plan for every disability by ourselves because we do not find ourselves disabled in every fashion imaginable. We don't know what we don't know – until people with a range of disabilities interact with our digital environments in different ways.

That's why testing with real users who rely on assistive technologies is invaluable. Automated accessibility tests can identify some coding errors, but they cannot replicate the lived experiences of users. For example, a visually impaired user might reveal navigation issues that aren't immediately obvious, such as a drop-down menu that disappears too quickly for a screen reader to detect.

Designing for assistive technologies also requires flexibility. Consider a video conferencing app. Providing captions for deaf users, keyboard shortcuts for those with motor impairments, and compatibility with screen readers ensures that the platform is accessible to a broad audience. This flexibility benefits everyone, making the app or website easier to use across different scenarios.

The impact of assistive technologies on design

When we design with assistive technologies in mind, we are not only solving problems, we are innovating. These tools push us to think beyond the average user and create experiences that are intuitive and equitable. By aligning design practices with the requirements of assistive technologies, we ensure that digital spaces are usable for all.

For instance, designing an e-commerce website with clear labels, scalable text, and logical workflows improves usability for screen reader users, but these same features also benefit someone shopping on a small screen or in a hurry. The principles of accessibility enhance the overall user experience, making the product better for everyone.

Assistive technologies provide essential access to digital content, but their effectiveness depends on how well they align with real user needs. Simply implementing these tools isn't enough. Designers and developers must consider how different disabilities shape interactions and ensure that accessibility features work in practical, meaningful ways. By mapping user needs to specific design solutions, we can move beyond generic fixes and create digital experiences that are intuitive, seamless, and empowering for all users.

Mapping user needs to design solutions

Designing for accessibility begins with understanding the diverse needs of users. The most effective solutions aren't one-size-fits-all – they're adaptable to different scenarios, disabilities, and assistive technologies. Meeting those needs starts by asking the right questions: Who are we designing for? What barriers might they encounter? How can we reduce or remove those barriers from the start?

One powerful approach is to build in customization. Letting users adjust font sizes, switch to dark mode, turn off motion, or choose their preferred input method puts control in the hands of the people who know their needs best. Flexible, user-tailored experiences are often the most accessible – and the most inclusive.

This process begins by asking critical questions:

- How would a user with limited mobility complete a purchase on an e-commerce site?
- What tools would a deaf student need to participate in a virtual lecture?
- How would a blind user independently navigate a transportation app to check real-time arrival times?
- What barriers might a user with ADHD encounter when filling out a complex online form?
- How can a person with dyslexia efficiently read and comprehend content on a news website?
- What accessibility challenges might arise for a user with a tremor who needs to interact with small touchscreen buttons?
- How does a user with color vision deficiency distinguish between success and error messages on a financial dashboard?
- What modifications would make a virtual reality experience accessible to users with various disabilities?

Each question uncovers unique challenges and opportunities for innovation.

To effectively design for accessibility, it's important to evaluate user needs systematically. This means identifying the types of disabilities a user might have, understanding the barriers they may encounter, and ensuring that digital content works with the assistive technologies users choose for themselves.

Mapping the user needs to design solutions involves recognizing those barriers and designing in a way that removes them – whether through semantic structure, keyboard navigation, clear visual design, or other inclusive strategies. Achieving this requires a deep understanding of the user journey and how various assistive technologies interact with digital environments.

Understanding the user journey

User journeys highlight how people interact with technology in real-world scenarios. By analyzing these journeys, we uncover friction points and areas for improvement. Consider the following example:

- **Scenario**: A student with dyslexia is using an online learning platform to complete a research paper.

- **Challenge**: The platform uses tightly spaced, justified text in a decorative serif font, and there's no way to adjust the visual presentation. These choices make it harder for the student to decode and retain information, increasing fatigue and frustration.
- **Solution**: Using a sans-serif, dyslexia-friendly font, adding ample line spacing, and offering customization options (such as changing text/background color contrast) can significantly reduce visual stress. While not a cure, optional text-to-speech integration may also support comprehension when decoding becomes too difficult.

By applying these principles and testing with neurodiverse users, you can gather valuable insights that highlight specific barriers and guide targeted accessibility improvements. By understanding the user's context, designers can prioritize changes that have the greatest impact.

Real-world applications of mapping needs

Understanding user needs is just the first step – applying those insights to real-world digital products is where accessibility makes a tangible impact. From education to e-commerce to financial services, different industries present unique challenges and opportunities for inclusive design. By examining common accessibility barriers in these spaces, we can identify practical solutions that improve usability for all users, not just those with disabilities. The following examples highlight how mapping user needs leads to more accessible and equitable digital experiences.

E-learning platforms: removing barriers to digital education

E-learning platforms often serve diverse users, including those with cognitive disabilities, visual impairments, and hearing loss. However, when accessibility is not considered, these platforms can create significant barriers to learning. For example, consider the following barriers and practical solutions:

Imagine this scenario: A college student who is blind, a second student who is deaf, and a third student with ADHD are enrolled in an online course that relies heavily on video content and lengthy reading assignments.

Now, what challenges might they face?

- Videos lack captions or transcripts, excluding d/Deaf and hard-of-hearing students
- Long, unstructured text overwhelms students with ADHD or dyslexia
- Blind students encounter navigation barriers and inaccessible quizzes that don't work with screen readers

What are the solutions?

- Provide captions and transcripts for all video content
- Allow adjustable playback speeds to support auditory processing needs
- Structure content with clear headings and concise paragraphs to enhance cognitive accessibility
- Ensure screen reader compatibility, including labeling all interactive elements
- Enable full keyboard navigation for quizzes and discussion forms

These changes improve access for students with disabilities while also benefiting multi-taskers, non-native speakers, and users in noisy or distracting environments.

While education opens doors to knowledge, accessible design in retail ensures those doors remain open when it's time to shop, connect, and simply take part in everyday life.

E-commerce websites: ensuring inclusive online shopping

E-commerce websites must cater to a wide range of abilities, yet many fail to account for accessibility, making online shopping frustrating or impossible for some users.

Imagine this scenario: A customer with low vision, another with limited hand mobility, and a third who uses a screen reader attempt to make purchases on a retail website.

What are the challenges?

- Form fields are visibly labeled but not programmatically labeled, confusing screen reader users
- Error messages signaled only by a change in font color aren't immediately interpretable by users with color vision deficiency
- Small touch targets and drop-down menus are difficult for users with motor impairments
- Product images with missing or non-descriptive alt text leave screen reader users without context

What are the solutions?

- Ensure form fields are programmatically (not just visually) labeled
- Supplement color-coded messages with icons or text
- Enlarge clickable areas and simplify interactions for keyboard and switch control users
- Add meaningful alt text to product images

These solutions expand usability for all shoppers while reducing cart abandonment and increasing sales fulfillment. But accessibility isn't just about convenience or conversion – it's about equity, especially when financial independence and personal security are at stake.

As more essential services move online, digital accessibility becomes a matter of civil rights, not preference. Tasks such as managing bank accounts, applying for disability benefits, signing medical forms, or reviewing legal documents must be both accessible and secure. For people with disabilities, especially those who use assistive technologies, privacy and autonomy are at risk when systems are not built with their needs in mind.

For example, a blind user completing a medical intake form via a screen reader should be able to do so independently – without needing a third party to read sensitive health information aloud. Likewise, deaf users must be able to receive important financial or legal information in formats that don't rely solely on audio.

Accessible design in these contexts must go beyond readability – it must prioritize accuracy, confidentiality, and user control. Inclusive systems aren't just good business practice; they are essential for protecting the dignity, privacy, and rights of disabled users in digital spaces where stakes are high.

Banking and financial tools: improving digital accessibility in finance

Banking apps and websites handle sensitive tasks that require precision and privacy. For users with disabilities, inaccessible design can create not just frustration, but a loss of autonomy – especially when it comes to managing finances independently.

Imagine this scenario: A blind user tries to enter their PIN but the field is not compatible with screen reader input. They're forced to ask someone else for help, compromising their privacy. A user with a cognitive disability can't rely on autocomplete or use a password manager due to poor implementation, increasing the chance of errors or security issues. A third user with limited mobility struggles to complete a transaction because the app requires mouse-only input.

What are the challenges?

- Visual CAPTCHAs, time-limited codes, and audio prompts that exclude blind, Deaf, or neurodivergent users
- Inaccessible PIN entry fields that prevent secure, independent authentication
- No support for autocomplete or password managers
- Dense financial jargon and cluttered data tables

- Layout distortion at 200% zoom, disrupting readability and navigation
- CAPTCHA systems without accessible alternatives

What are the solutions?

- Implement accessible login options such as biometrics and multi-factor authentication that don't rely on vision or sound
- Ensure PIN fields and secure forms are compatible with screen readers and accessible input methods
- Support autocomplete, saved credentials, and password manager integration
- Reformat financial data using plain language, logical grouping, and accessible data tables with headers, concise labels, and screen reader–friendly markup
- Test for compatibility with screen magnifiers and browser zoom (especially at 200%) to catch responsive design issues early
- Offer non-visual CAPTCHA alternatives, such as logic-based or multi-choice options

These enhancements go beyond technical compliance. They protect privacy, support independence, and build trust – all essential in the financial space. Accessibility in banking isn't just about inclusion – it's about dignity, security, and equal access to financial empowerment.

Iteration: a key to success

Mapping user needs is not a one-time exercise but an ongoing process. As user needs evolve, so must the solutions designed to meet them. Consider a company that rolls out an accessible version of its app only to discover through feedback that its live chat feature is incompatible with screen readers. This feedback loop is vital to refining designs and maintaining inclusivity.

Iteration involves collaboration between designers, developers, and users. Conducting regular usability tests with diverse participants ensures that solutions remain relevant and effective. For instance, testing a new search filter with users who rely on keyboard navigation might reveal hidden barriers, prompting improvements before the feature is widely released.

Identifying gaps and bridging them

No product is perfect on the first attempt. Gaps often exist between user needs and the capabilities of current technologies or designs. Recognizing these gaps is the first step toward bridging them.

For example, consider a virtual event platform designed to host webinars. If the platform offers automatic captions but fails to provide manual editing options, the captions (or "craptions"!) may be riddled with inaccuracies. The gap here is the lack of a feature to ensure caption accuracy, which could exclude deaf attendees or those with auditory processing challenges.

Bridging gaps involves prioritizing enhancements that align with user feedback. This iterative mindset improves accessibility, drives innovation, and elevates the overall user experience.

Effective accessibility solutions require more than a one-time fix; they demand continuous evaluation, iteration, and user feedback. By mapping user needs, implementing assistive technologies, and refining digital experiences over time, we create systems that are not just functional but truly inclusive. However, understanding accessibility in theory is different from experiencing barriers firsthand. The next section explores how centering lived user experiences can deepen empathy, highlight unseen obstacles, and lead to more effective, user-centered design solutions.

Centering lived experience: designing with, not for

Empathy is often cited as a foundation of inclusive design, but without direction, empathy alone can become a dead end. You cannot fully understand what it means to be disabled by *pretending* to be disabled. Instead of stepping into someone else's shoes, inclusive design calls on teams to step into shared work – bringing disabled people into the process, listening deeply, and acting on what they share.

Accessibility done well isn't about imagining barriers – it's about removing real ones, in partnership with the people most affected by them.

From observation to collaboration: building insight through lived experience

Lived experience is not anecdotal – it's expertise. Working directly with disabled people throughout the product life cycle leads to insights no simulation could ever provide.

Instead of guessing what a user might struggle with, invite them in:

- Recruit disabled users as part of your usability testing pool, ensuring you include a range of disabilities, assistive technologies, and intersectional perspectives.
- Pay disabled consultants, advisors, and testers equitably for their time and expertise. Accessibility should never rely on unpaid emotional or educational labor.
- Review personas and user journeys with disabled team members or advocates to ensure your assumptions reflect reality – not stereotypes.

Whether you're building a new product or retrofitting an existing one, the most effective feedback will come from people who use the technology every day, not from internal teams trying to simulate unfamiliar experiences.

Beginner tip: Start with moderated usability testing sessions with 3–5 participants who use different assistive technologies (e.g., a screen reader user, a switch device user, and someone who relies on voice control).

Expert-level move: Establish a standing disability advisory council, or embed disabled professionals into product and UX teams from the outset.

Move from testing to partnership: integrating feedback into workflows

Feedback isn't valuable unless it's integrated into decision-making. One-time user testing is a start, but sustainable accessibility comes from building feedback loops into every sprint, release, and review.

Here's how to build accessible collaboration into your process:

- Use real assistive tech (not simulations) during QA and **internal** reviews. Test with screen readers (JAWS, NVDA, VoiceOver), screen magnifiers, switch controls, and voice navigation tools.
- Create bug-reporting or barrier-tracking mechanisms that include severity ratings and accessibility impact, so fixes are prioritized alongside visual or functional defects.
- Bring designers, developers, and QA into user debriefs, not just product owners. Shared insight leads to shared accountability.
- Document what was learned and share it across teams. Avoid siloed knowledge that leaves accessibility up to one champion.

Beginner tip: Add a screen reader test to your Definition of Done.

Expert-level move: Assign accessibility liaisons across product teams who connect directly with disabled users to close the feedback loop.

From awareness to action: creating a culture of accountability

True inclusion doesn't stop at testing – it lives in your organizational values, your workflows, and your hiring practices. Creating a culture where accessibility is expected, not exceptional, requires ongoing commitment:

- Replace empathy workshops or simulations with training led by disabled people, focused on real-world impacts and solutions.
- Celebrate accessibility wins and highlight lived experience voices during sprint demos, town halls, or retrospectives. Visibility matters.
- Ensure leadership supports accessibility as a cross-functional priority, not an afterthought owned only by compliance teams.
- Measure progress using metrics such as usability scores from disabled users, not just WCAG pass/fail rates.

> Beginner tip: Launch an internal accessibility education series that spotlights disabled voices and practical tools.
>
> Expert-level move: Tie accessibility metrics into OKRs or KPIs, aligning inclusion with business goals.

You don't need to simulate disability to understand the need for inclusion. You need to listen, hire, involve, and invest. Disabled people don't need to be imagined – they need to be heard, trusted, and included. Their insights aren't edge cases – they're central to building systems that work for everyone. When accessibility is informed by real experiences, the result is better products, better outcomes, and a more just digital world.

Summary

In this chapter, you explored how different types of disabilities – visual, auditory, motor, cognitive, and mental health – shape the way users interact with digital experiences. You learned how these differences influence access needs and how various assistive technologies, such as screen readers, refreshable Braille displays, switch controls, and magnifiers, support access for millions of users. We also looked at the importance of clear content structure, meaningful labels, keyboard navigation, and flexible interfaces in supporting usability across disability types.

Understanding these user needs is essential for creating accessible products that go beyond technical compliance. You saw how inaccessible design choices can limit access to critical services such as banking and healthcare, and how inclusive practices – such as plain language, accessible authentication, and support for customization – can restore autonomy and trust. This chapter also emphasized the importance of testing with real users and assistive technologies, and of centering lived experience rather than relying on simulations.

In the next chapter, we'll move from understanding user needs to applying legal and ethical considerations to product design. You'll learn how to align your work with accessibility laws and standards, integrate inclusive practices into design systems and development workflows, and scale accessibility across teams and organizations.

3

Legal and Ethical Considerations in Accessible Design

By Maya Sellon

When you turn on captions to watch your favorite program on Netflix, you're benefiting from the results of accessibility laws in action. These legal protections don't just ensure compliance; they drive the creation of more inclusive experiences for everyone. But accessibility isn't just about compliance – it's also about ethics. A product that merely meets the legal minimum might still be unusable for many, whereas an ethically driven approach ensures an experience that feels as seamless as magic. Inclusive design should be baked into the process, not sprinkled on top.

Certain regulatory frameworks, such as the **US Americans with Disabilities Act** (**ADA**) and the **UN Convention on the Rights of Persons with Disabilities** (**CRPD**), have established legal requirements for accessibility and set the foundation for ensuring digital spaces are inclusive. Yet, laws alone are only the starting point. True inclusion requires going beyond compliance. Ethical design shifts the focus from meeting minimum standards to creating inclusive experiences. It recognizes that barriers are created by design choices, not by individuals with disabilities.

By the end of this chapter, you'll gain a clear awareness of the major accessibility laws and international protections that establish accessibility as a fundamental human rights issue, not a passing trend or design preference. With this foundation, you'll be able to confidently push back against resistance to accessibility measures, whether it's questions about user demographics or cost-driven business concerns. You'll also better understand how legal requirements and ethical considerations intersect, equipping you to advocate for inclusive design as a core principle rather than an afterthought.

In this chapter, we will cover the following topics:

- Navigating global accessibility legislation
- Balancing legal compliance and ethical responsibility
- Forecasting future legal trends in digital accessibility
- Cultivating an ethically-driven design culture

Navigating global accessibility legislation

Legal protections for disability rights are more than just regulatory requirements. They define accessibility as a fundamental human right and shape how societies approach inclusion. While accessibility laws differ by country, they serve as a critical tool for ensuring equal access, preventing discrimination, and holding organizations accountable. However, legal protections are far from universal. Only 24% of countries have constitutions that specifically prohibit discrimination or affirm equal rights based on disability, and less than 10% of country constitutions explicitly guarantee civil rights for persons with disabilities (UCLA WORLD Policy Analysis Center, 2024). Constitutional protections serve as a crucial starting point for challenging inequality and dismantling discriminatory laws. In Uganda, legal reforms have led to mandatory workplace accommodations for people with disabilities (Shakespeare et al., 2019), while in Peru and Colombia, constitutional guarantees have secured the right to vote for persons with disabilities (Open Society Foundations, 2019). While these successes highlight what's possible, significant barriers remain – something we'll explore later in this chapter.

Beyond setting legal obligations, these laws influence broader societal attitudes. The adoption of the social model of disability, championed by the UN CRPD, has reframed disability as an issue of external barriers rather than individual impairment. This shift has had far-reaching consequences, encouraging policies and design practices that prioritize accessibility from the outset rather than treating it as an afterthought.

Here is why legal protections matter globally:

- **Human rights and equality**: Disability rights are human rights. More than 90% of the United Nations member countries ratified the CRPD because they recognized that disability discrimination was a global issue that needed to be addressed.
- **Combating systemic discrimination**: Laws protect against exclusion and help break down barriers in employment, education, and public services, ensuring that people with disabilities aren't faced with systemic barriers that limit their opportunities and independence.

- **Promoting global alignment**: International laws encourage countries to align their accessibility efforts, providing developing nations with a foundation to build on and supporting cooperation across borders (United Nations Department of Economic and Social Affairs [DESA], n.d.-b).
- **Accountability and enforcement**: Legal agreements like the CRPD require governments to track progress on disability rights, helping to push for stronger enforcement and preventing accessibility efforts from being ignored.
- **Economic impact**: It's good for business. Reducing the employment gap between people with and without disabilities enhances economic inclusion by enabling more people to contribute productively to the economy, which in turn boosts GDP and promotes economic inclusion (Halder & Argyropoulous, 2019).

Legal protections shape accessibility rights, but their strength lies in the systems that uphold them. Over the years, landmark laws have influenced policies and standards worldwide, providing a roadmap for accessible digital environments.

Establishing the legal foundation for digital accessibility

The legal recognition of accessibility as a human right has evolved through key global movements and landmark legislation. These milestones shaped the frameworks that influence digital accessibility today.

The International Year of Disabled Persons marked a turning point in how disability was framed, emphasizing that disability results from the interaction between an individual and the barriers in their environment. This shift led to the World Programme of Action Concerning Disabled Persons, which established a global agenda for inclusion, and later to the Standard Rules on the Equalization of Opportunities for Persons with Disabilities (United Nations General Assembly [UNGA], 1994). These frameworks played a key role in influencing the development of policies and actions worldwide and are formally recognized by the UN CRPD as important foundations for advancing equal opportunities for persons with disabilities (UNGA, 2006).

In the United States, the Rehabilitation Act of 1973, particularly Section 504, became a landmark U.S. law prohibiting discrimination based on disability in federally funded programs. This was later extended through Section 508 (1998), which required federal agencies to make their electronic and information technology accessible, aligning with early WCAG principles (U.S. Congress, 2016).

The US Americans with Disabilities Act introduced a civil rights approach to disability, promoting inclusion, accessibility, and equal status. These principles influenced the development of international frameworks, particularly the UN CRPD, which adopted a similar rights-based model for advancing disability rights globally (Blanck, 2023; Kanter, 2003; Szymanski, 2009).

These developments, along with growing international advocacy, led to the drafting of the UN CRPD in 2006 – the first comprehensive human rights treaty of the 21st century, and a pivotal step in recognizing accessibility, including digital, as a fundamental human right (UN DESA, n.d.-a). Article 9 requires states to ensure access to the physical environment, transportation, information, and communications, including digital services, underscoring the importance of removing barriers across all areas of life (UNGA, 2006).

These legal foundations explain why accessibility isn't just good design – it's a legal right. Knowing where these obligations come from equips you to advocate for inclusive, compliant digital products with confidence grounded in global legal standards.

Digital accessibility and global legal frameworks

Digital accessibility has become a critical focus in affirming the rights of people with disabilities and eliminating systemic discrimination. The **Web Content Accessibility Guidelines** (**WCAG**) is an internationally recognized standard for digital accessibility, with WCAG 2.0 also formalized as ISO/IEC 40500:2012. ISO/IEC 40500:2012 is expected to be replaced by ISO/IEC DIS 40500, aligning with WCAG 2.2 (International Organization for Standardization, 2012).

WCAG defines three levels of conformance – A, AA, and AAA – which indicate increasing levels of accessibility. Each level builds on the previous one, meaning Level AA includes all Level A requirements, and Level AAA includes all of A and AA, plus additional success criteria. Level AA is the most commonly adopted standard in legislation and policy (Rogers, 2025). While Level AAA represents the highest standard, even content at this level may not fully address the diverse types, degrees, or combinations of accessibility needs, particularly in cognitive, language, and learning areas (World Wide Web Consortium [W3C], 2024b).

These guidelines outline measurable success criteria for evaluating whether websites, apps, and digital content are usable by people with disabilities. Although WCAG itself is not legally binding, it often forms the basis for legislation when adopted into national or regional laws.

For example, the **European Accessibility Act** (**EAA**), set to take effect in 2025, is a major step toward ensuring accessibility across EU member states. Each EU country is responsible for implementing the EAA's requirements, determining penalties for non-compliance, and designating enforcement bodies, though some countries already have stronger accessibility laws in place. The EAA introduces common requirements for specified digital products and services, reinforcing WCAG as a benchmark through its reference in EN 301 549, the European technical standard for digital accessibility. This standard establishes baseline accessibility requirements across the EU, aligning with WCAG principles and the broader goals of the UN CRPD (European Parliament, & Council of the European Union, 2019; European Telecommunications Standards Institute, 2021).

In April 2024, the U.S. Department of Justice reinforced the importance of digital accessibility, formalizing WCAG 2.1 Level AA as the legal standard for public entities under the Americans with Disabilities Act (U.S. Department of Justice, 2024). While the U.S. legal landscape is shaped by a mix of federal, state, and local laws, with court rulings often leading to fragmented interpretations, this decision further cements WCAG as a foundational standard for web accessibility.

The influence of WCAG extends beyond the U.S. and Europe, shaping digital accessibility laws worldwide. Countries such as New Zealand, Moldova, and Zimbabwe have adopted WCAG into their national design standards (Jordanoski & Meyerhoff Nielsen, 2023). Brazil's Law for Inclusion of Persons with Disabilities (Law 13.146, 2015) mandates that both public and private sector websites meet accessibility standards (Bianchi, 2021). These laws reflect a growing international commitment to ensuring that digital spaces are inclusive and accessible and highlight WCAG's pivotal role in shaping accessibility standards worldwide.

One of the earliest global legal victories for digital accessibility came in 2000, when Bruce Maguire successfully challenged the Sydney Organising Committee for the Olympic Games (SOCOG) under Australia's Disability Discrimination Act for failing to make their website accessible for him as a blind user to purchase tickets online. This ruling marked a landmark moment, establishing that digital inaccessibility could constitute unlawful discrimination (Russell, 2003).

While international laws and accessibility standards set expectations, their effectiveness often depends on enforcement. Legal action plays a key role in ensuring compliance, holding organizations accountable when accessibility is overlooked. One landmark case, National Association of the Deaf, et al. v. Netflix, resulted in a historic settlement requiring Netflix to provide 100% closed captions on its on-demand streaming content within two years (Disability Rights Education & Defense Fund, 2012). The Netflix lawsuit was a pivotal moment for digital accessibility, setting a legal precedent that influenced accessibility policies across the streaming industry, requiring content providers to prioritize inclusive design from the start. This is why, today, you can turn on captions and enjoy almost any program on Netflix – an accessibility victory made possible through legal action.

The challenges that remain

The Netflix lawsuit underscored the importance of accessibility while revealing a deeper issue: laws are only effective when they are enforced. Legal victories like this help drive industry-wide change, but enforcement remains inconsistent across regions and industries. Yet, despite legal progress, gaps in enforcement, implementation, and awareness mean that barriers persist. Below are some of the most pressing challenges that continue to hinder accessibility progress:

- **A lack of awareness and education**: Many organizations remain unaware of their duty regarding legal accessibility requirements, leading to poor implementation and widespread noncompliance.
- **Limited enforcement mechanisms**: Even where accessibility laws exist, enforcement is inconsistent, leading to non-compliance.
- **Resource constraints**: Small businesses and underfunded organizations may struggle to meet accessibility standards without financial or technical support.
- **Rapid technological changes**: Laws and standards struggle to keep pace with evolving digital technologies, creating gaps in compliance requirements.

However, a country's or an organization's wealth alone doesn't dictate accessibility success – focused strategies, education, enforcement, and targeted awareness campaigns to address stigma and cultural attitudes are crucial. Challenges will persist without proper support, even with the adoption of legal frameworks and standards (Jordanoski & Meyerhoff Nielsen, 2023).

These challenges aren't limited to organizations. They reflect broader global patterns. While the CRPD sets a powerful benchmark, not all countries have committed equally, and many that have still struggle to meet their obligations. According to the United Nations Disability and Development Report (2024), accessibility efforts, including in digital spaces, need to accelerate up to 65 times faster to meet global inclusion goals by 2030.

This highlights an important reality: legal frameworks lay the groundwork for accessibility, but compliance alone does not guarantee an inclusive experience. Even in regions with strong laws, organizations must move beyond only meeting legal requirements and embrace their ethical responsibility to create meaningful, lasting change.

Balancing legal compliance and ethical responsibility

For many businesses, legal compliance is the primary motivator for accessibility initiatives. Laws such as the ADA, EAA, and the UN CRPD provide a clear legal foundation. The risk of lawsuits, financial penalties, and reputational harm offers compelling arguments to prioritize accessibility. However, a compliance-first mindset risks turning accessibility into a tick-box exercise, where meeting minimum requirements overshadows the broader ethical responsibility to design inclusive experiences that work for everyone.

So, how do you ensure accessibility doesn't become just another box to tick?

It starts with a mindset – one that views accessibility as part of a longer journey, not a quick fix. If you're working to improve accessibility, you're helping lead a cultural transformation, one that prioritizes real inclusion over speed or surface-level solutions. Tools can help, but only when guided by a clear understanding of what they can – and can't – do.

The role and limitations of automated solutions

Automated testing uses software tools to quickly scan digital content against accessibility standards. These tools are particularly valuable for their speed, scalability, and efficiency, supporting high-volume reviews and identification of technical issues. A study by Deque, a web accessibility software and services company known for its axe suite of testing tools, which analyzed over 13,000 web pages, found that automated tools can detect just over 57% of accessibility issues present on a web page (Deque Systems, n.d.). This makes automated testing invaluable for identifying common, quantifiable errors such as missing alternative text or missing form labels.

However, automated testing's strengths come with significant limitations. The same study revealed that automated tools could evaluate only 16 of the 50 success criteria defined under WCAG 2.1 Level AA, representing approximately 30% of the total criteria (Deque Systems, n.d.). This subset focuses on the most frequent and easily measurable accessibility issues but leaves 70% of success criteria requiring manual evaluation by accessibility professionals or through user testing to ensure compliance and usability.

This distinction is critical. Automated tools may excel at identifying quantifiable technical issues, but they currently cannot evaluate the quality or contextual aspects of accessibility solutions. This gap can create false assurances, giving the illusion that a digital product meets accessibility requirements when, in practice, it does not provide a genuinely inclusive experience.

For example, an automated tool might confirm the presence of alternative text, but it does not evaluate whether the alternative text is meaningful or conveys the purpose of the image. Perhaps an image contains critical information, such as a support phone number, and the alternative text has been set to “call center support number” but omits the actual number. An automated tool would likely not flag this as an error, but the phone number remains inaccessible to screen reader users.

These kinds of gaps in automated testing reveal a broader challenge with relying too heavily on automated approaches to accessibility. As the accessibility landscape evolves, the market for tools is expanding beyond detection into automated solutions. While automated testing helps detect some accessibility issues, other tools are now being marketed as automated fixes. Overlays, for example, are third-party tools that claim to resolve accessibility problems without requiring changes to the underlying code. They are often positioned as cost-saving options that allow product teams to avoid training or rework costs; the overlay handles the accessibility while the team continues to work as usual.

However, these tools are not without risk, and relying on them could lead to legal and reputational harm. In one high-profile case, the U.S. Federal Trade Commission fined a company $1 million in 2025 for misleading advertising, ruling that it had violated federal law by misrepresenting the capabilities of its overlay (Law Office of Lainey Feingold, 2025). Critical evaluation is essential, and promises of “easy fixes” should be approached with caution.

Ethical dimensions of accessible design

Legal compliance sets a baseline for accessibility, but organizations must also consider the ethical implications of their designs. For instance, thoughtfully designed alternatives, such as text versions of complex visuals, can enhance accessibility and inclusivity when considered from the outset. However, offering an “accessible version” of a digital product or website may technically meet accessibility standards but risk marginalizing users by reinforcing a sense of otherness. Creating bolt-on versions for users with disabilities can echo historical practices of segregation and fail to reflect the principles of universal design.

Instead, ethical responsibility demands that organizations prioritize inclusivity from the outset. Designing for accessibility from the beginning not only ensures compliance but also fosters user experiences that are equitable and respectful. It also aligns with business objectives by broadening market reach and enhancing brand reputation.

For example, motion design, such as animations and transitions that add polish to modern interfaces, can enhance visual appeal but may overwhelm users sensitive to motion, such as those with vestibular disorders. Thoughtful design anticipates this need by respecting user preferences. Companies like Apple and Microsoft enable users to set "reduce motion" preferences at the system level. It is important to respect these preferences, never override them, and ensure alternative experiences are thoughtfully created to preserve clarity and usability while maintaining the overall design integrity.

Similarly, in the gaming industry, accessibility designed from the outset has demonstrated the power of inclusive approaches. Retrofitting a game to add basic keyboard controls might meet minimum compliance requirements, but often results in an inferior user experience. In contrast, games built with accessibility in mind have allowed blind players to navigate racetracks independently using aural feedback. This approach is about creating inclusive experiences, not merely meeting a requirement. It shows how designing inclusively can drive innovation, unlock creativity, and extend market reach.

Moving beyond a tick-box exercise

Accessibility is not simply about avoiding lawsuits; it is about recognizing that the barriers individuals face are not inherent to them but are created by the systems, tools, and environments we design. Providing a separate, "accessible version" of content might technically meet legal standards, but it feels like an afterthought. It comes across as a hastily prepared option that says, "We forgot about you, but here's something that'll have to do."

To bridge the gap between compliance and usability, organizations must shift their focus from merely meeting requirements to creating accessible and inclusive products. This shift involves:

- **Investing in robust monitoring systems**: Accessibility is not a one-time fix. Ongoing monitoring tools keep accessibility visible and top of mind, supporting ongoing commitment even as teams change and digital products evolve.
- **Embedding accessibility into the product development lifecycle**: Accessibility must be embedded across design, development, and testing. When designers, developers, and testers approach it as part of their job, it stops being an extra task, reduces unnecessary rework, and simply becomes how things are done.
- **Shifting leadership attitudes**: Without leadership buy-in, accessibility remains underfunded, deprioritized, and reactive. Many organizations fall into a self-fulfilling cycle where accessibility always feels like an expensive, last-minute burden.

Think of inclusive design as being like baking blueberry muffins. When accessibility is mixed in from the start, like blueberries folded into the batter before baking, it becomes an integral part of the experience. The flavors are absorbed, the batter takes on the richness of the blueberries, and the result is a cohesive, thoughtful product that delights everyone. In contrast, retrofitting accessibility after the fact is like trying to shove blueberries into plain muffins after they've come out of the oven: it's messy, unappealing, and ultimately fails to deliver the same experience.

As accessibility auditing becomes increasingly sophisticated, it is vital to remember that tools and processes are only as effective as the intent behind them. Legal compliance is a necessary foundation, but ethical responsibility ensures that accessibility efforts translate into meaningful and equitable experiences for all users.

Likewise, as technology evolves and digital landscapes shift, accessibility laws must keep pace with innovations like AI-driven interfaces and the increasing reliance on digital public services. To better anticipate these changes, let's explore how emerging legal trends are shaping the future of accessibility.

Forecasting future legal trends in digital accessibility

As technology advances and digital services become more embedded in everyday life, accessibility laws must evolve to keep pace. Several emerging trends are shaping how legal frameworks address accessibility, from increasing reliance on digital services to the role of artificial intelligence. Understanding these trends can help individuals and organizations anticipate future legal requirements and proactively integrate accessibility into their strategies.

The increasing reliance on digital services

Governments and organizations worldwide are rapidly digitizing services, from healthcare to banking to education. For example, the rise of digital IDs, e-visas, and online public services offers greater convenience but also risks widening accessibility gaps if not designed inclusively. However, this progress depends on critical prerequisites such as reliable devices, internet connectivity, and affordable utilities like electricity. Without addressing these foundational needs, digital-first strategies risk substituting traditional barriers with new ones, particularly for socioeconomically disadvantaged individuals and communities.

Legal frameworks are beginning to respond to these challenges. The European Accessibility Act, for example, requires many essential digital services to be accessible by 2025, setting a new precedent for accessibility enforcement across industries. However, compliance remains inconsistent globally, and organizations must prepare for stronger regulations like the EAA requiring accessible digital infrastructure in the coming years.

What to keep in mind:

- Design for the real-world population, not just ideal conditions. Not everyone will have the latest device, the fastest internet connection, or even consistent access to broadband. Use demographic data, device usage statistics, and average internet speeds to ensure your service works for the people who actually need it, not just those with cutting-edge tech.
- Provide alternative access methods. Some users may prefer or require non-digital options, whether due to age, disability, or personal preference. Ensure that essential services aren't locked behind online-only platforms without accessible fallback options, such as phone support or in-person assistance.
- Consider digital literacy and discoverability. Even if users can get online, they need to know where to go and understand how to use your service. Complex sign-up processes, unclear navigation, or overly technical language can be just as much of a barrier as a missing screen reader label.

At the same time, societal awareness, driven by social media platforms, public voices, and cultural representation, fosters a broader normalization of disabilities.

Cultural shifts as a predictor for policy change

Authentic representation plays an important role in breaking down stereotypes and reshaping public attitudes toward disability. For example, Breaking Bad actor RJ Mitte, who has cerebral palsy, played a character with the same condition. And Marissa Bode, who uses a wheelchair, was the first to be authentically cast as Nessarose in the recent Wicked film adaptation. Beyond visibility, these portrayals provide an opportunity to shape how disability is represented, ensuring that stories reflect real experiences rather than relying on outdated tropes.

Social media has also amplified conversations around disability and neurodivergence, making these conditions more visible and relatable to the public. This increased visibility creates a ripple effect, paving the way for the conscious inclusion of people with disabilities throughout policy-making processes. For example, the draft WCAG 3.0 standard explicitly states its aim to address a broader range of user needs, including cognitive disabilities (W3C, 2024a), a sign of how societal shifts are reflected in policy evolution.

As cultural shifts continue to shape accessibility expectations, organizations and policymakers alike must consider how these changes influence public attitudes and legal priorities.

What to keep in mind:

- Public expectations shape corporate responsibility. Representation alone is not enough. As disability inclusion becomes more visible, customers, employees, and stakeholders will expect companies to deliver on accessibility, not just market it. Businesses that fail to keep up risk losing credibility and consumer trust.
- Representation should extend beyond marketing. Disability inclusion is about more than showcasing diversity in campaigns – it requires hiring, consulting, and working with disabled people throughout the design process. Assumptions and stereotypes thrive when people with lived experience aren't involved in decision-making, leading to products, services, and policies that don't meet real needs.

As accessibility expectations continue to rise, another key factor influencing accessibility policies is the world's ageing population. As people live longer and remain in the workforce later in life, accessibility will need to expand beyond disability inclusion and account for age-related changes in vision, hearing, mobility, and cognitive function.

The impact of an ageing population

The global population is ageing. In 2022, 770 million people aged 65 or over made up 9.7% of the world's 7.9 billion population. By 2050, the UN projects this number to rise to 1.6 billion, or 16.4% of a population of 9.7 billion (UN DESA, 2022). This demographic shift will significantly impact workplace accessibility, healthcare services, and digital platforms, requiring organizations to design for age-related changes in vision, hearing, mobility, and cognition.

Many of these challenges overlap with existing standards, such as guidelines for text readability, high contrast, and keyboard navigation. However, most accessibility policies have not yet fully adapted to address the specific needs of an ageing workforce and consumer base.

Governments and economic policy groups are beginning to respond. The World Economic Forum's longevity economy principles and the G7's Solfagnano Charter highlight the importance of involving individuals with disabilities in policymaking and designing systems that support healthy ageing, equitable employment, and social inclusion (G7 Italia 2024, 2024). As populations continue to age, organizations must anticipate stronger accessibility regulations aimed at improving usability for older adults. The ongoing transposition of the EAA into national legislation by EU member states demonstrates how these principles can translate into actionable legal progress and provide a clear model that bears watching.

What to keep in mind:

- Age-related accessibility needs go beyond disability inclusion. Older adults experience gradual changes in vision, hearing, dexterity, and cognitive function, meaning design must account for clear contrast, larger text, and intuitive interfaces. WCAG success criteria – including plain language, predictable navigation, and larger target sizes for interactive elements – can help address these needs.
- Focusing only on public-facing accessibility while neglecting your own employees is a costly mistake. As people stay in the workforce longer, failing to provide accessible tools and environments will lead to decreased productivity, disengagement, and ultimately higher costs for businesses. Organizations should ensure that internal platforms, digital tools, and workplace policies meet accessibility standards, not just their customer-facing services.

As accessibility requirements expand, emerging technologies such as artificial intelligence present both opportunities and risks in shaping the future of digital accessibility.

Addressing accessibility in AI governance

Artificial intelligence has the potential to significantly improve accessibility, but when not designed with diverse users in mind, it can also exacerbate existing barriers. AI-driven features such as speech recognition and real-time generated image descriptions have transformed access for many people with disabilities. At the same time, AI frequently excludes or perpetuates harmful stereotypes when trained on limited datasets or developed without human oversight.

For example, speech recognition technology often struggles with non-standard speech patterns, strong accents, or speech impairments, making voice assistants unreliable for some users. AI-generated content can also reinforce stereotypes, by producing images including wheelchairs or depicting all people with hearing aids to be elderly, reflecting the biases embedded in the data it was trained on.

Governments are beginning to respond to these challenges. The European Union's AI Act and the United States' AI Bill of Rights reflect a growing consciousness of the ethical implications of artificial intelligence and the need for safe, inclusive, and transparent system designs. These frameworks align closely with accessibility goals and acknowledge AI's potential to create or eliminate barriers in critical areas such as healthcare, education, and employment (European Commission, 2021; White House, n.d.). As AI adoption continues to accelerate, integrating accessibility requirements into AI governance must play a significant role in shaping future legislation.

What to keep in mind:

- AI is only as inclusive as the data it learns from. When selecting AI models, organizations should examine the datasets used for training to ensure they represent the full diversity of human experiences, including disability, neurodivergence, and non-standard speech patterns. If developing custom AI, teams must actively include diverse inputs to prevent bias from excluding or misrepresenting users.
- AI-generated accessibility features should be treated as a supplement, not a replacement. Automated captions, alt text, and other AI-driven solutions can enhance accessibility but should not be solely relied upon. Human review remains essential to ensure accuracy, usability, and contextually meaningful output.
- AI accessibility is becoming a regulatory issue, not just an ethical one. As AI adoption expands, companies will be held accountable for ensuring that automated systems do not create new accessibility barriers. Staying informed on global AI accessibility regulations will be key to maintaining compliance.

By observing societal movements, increasing authentic media representation, and tracking public discourse – whether in evolving attitudes toward ageing populations, disability inclusion, or AI regulation – it becomes clear that accessibility laws do not exist in isolation. They are shaped by the shifting needs of society. These changes highlight the growing expectation that accessibility is not simply a compliance issue but a fundamental part of digital and societal progress.

As these movements continue to gain momentum, individuals and organizations must remain engaged, ensuring that inclusion is not an afterthought but a guiding principle. This means embedding accessibility into the very culture of how organizations design, build, and innovate.

Cultivating an ethically driven design culture

As digital accessibility evolves, organizations must foster a design culture that treats inclusivity as an essential part of the creative process. By embedding awareness, education, and ethical responsibility into their workflows, teams can ensure they create solutions that work for everyone.

Awareness as the catalyst for change

Accessibility gaps frequently stem not from intentional exclusion but from a lack of awareness about the diversity of user needs. Designers often focus on aesthetics and functionality without realizing how decisions, such as color contrast, screen layout, or visual effects, can impact usability for individuals with disabilities or anyone facing situational challenges. For instance, someone with low vision may rely on zooming or dark mode to reduce eye strain, but these features also benefit users working in dim environments or experiencing fatigue after a long day.

Raising awareness is the first step toward designing inclusively. Designers craft experiences for everyone, whether it's an older user adjusting font sizes for comfort, a professional toggling dark mode to reduce eye fatigue, or a friend relying on captions in a noisy café. When design decisions reflect real-world use cases, they move beyond aesthetics and ensure everyone's needs are considered.

Education as a foundation for action

Awareness alone is not enough. Teams need actionable knowledge to integrate accessibility into their workflows. Standards like WCAG (Web Content Accessibility Guidelines) provide a reliable foundation, offering measurable criteria to ensure consistent, inclusive design. These standards act as essential guideposts, providing clarity and confidence that a product is usable by everyone.

However, accessibility education cannot rest solely on designers. It must be embedded across entire teams, from developers and content creators to testers and leadership. Continued education through certifications, workshops, webinars, or support from accessibility specialists ensures teams remain current with emerging standards, legal developments, and evolving technologies. This shared knowledge also reduces barriers to collaboration, streamlines decision-making, and prevents accessibility from being treated as an afterthought.

Investing in education – specifically *ongoing* education – signals that accessibility is not optional; it is integral to high-quality, user-centered design.

Embedding accessibility into the design process

Building upon team awareness, an ethically driven design culture integrates accessibility into every stage of the design process, from research and prototyping to testing and iteration. By identifying and addressing barriers early, teams can create more inclusive and thoughtful experiences while avoiding the costly need to retrofit accessibility after the fact.

Designers play a critical role in this process. Incorporating accessibility reviews into daily workflows, whether by testing color contrast, ensuring logical navigation for keyboard and screen reader users, or considering the impact of visual elements like animations, helps embed accessibility into the core of a design.

Design culture, by its nature, is never static. It thrives on improvement, iteration, and problem-solving. An ethically driven culture embraces accessibility as a continuous process, applying the same iterative, problem-solving mindset used to improve other design outcomes. This means accessibility must be proactive, not reactive – something designers build in, not bolt on.

Ultimately, designers are not just building products; they are shaping experiences. Experiences can be welcoming, thoughtful, and empowering for every user. Or, they can exclude, frustrate, and undermine their efforts. By recognizing their influential role and embracing accessibility as a hallmark of great design, designers can harness their creativity to drive meaningful, beautiful, and lasting change.

Accessibility cannot rest on one individual or team alone. Collaboration across design, development, content, and quality assurance ensures that inclusive principles are embedded throughout a product's lifecycle, creating cohesive and effective solutions for all users.

Summary

This chapter explored the intersection of legal compliance and ethical responsibility in accessibility. While laws such as the ADA, CRPD, and EAA provide a legal foundation, true inclusion requires more than just meeting legal requirements. The hallmark of great design is embedding accessibility into culture, processes, and decision-making – because accessibility is not just about rules and regulations, it's about people.

Whether through AI-driven technologies, shifting societal expectations, or the reality of an ageing population, accessibility will impact everyone at some stage in their life. By failing to prioritize it early, organizations risk creating barriers that exclude users, increase costs, and diminish long-term sustainability. But by embracing accessibility as a design principle rather than an afterthought, businesses and individuals alike can drive meaningful, lasting progress.

We examined the limitations of automated testing, the impact of cultural and technological shifts on accessibility policies, and the importance of embedding accessibility into organizational culture. By understanding these factors, individuals and businesses can anticipate legal trends, advocate for ethical design, and implement inclusion as a core principle, not just a legal requirement.

Embedding accessibility into culture and processes is what transforms compliance into meaningful inclusion. But knowing the importance of accessibility isn't enough – you need to ensure that the people who will use your products are actually part of the design process. In the next chapter, we'll dive into how user research and testing help create accessible digital products that meet real-world user needs, not assumptions.

References

Navigating global accessibility legislation

Bianchi, F. C. (2021). *Digital accessibility innovation in Latin America and the Caribbean*. In M. A. Stein & J. Lazar (Eds.), *Accessible technology and the developing world* (pp. 81–105). Oxford University Press. `https://doi.org/10.1093/oso/9780198846413.003.0005`

Blanck, P. (2023). *International disability law and policy* (pp. 148–152). Edward Elgar Publishing. `https://doi.org/10.4337/9781802208337.ch11`

Disability Rights Education & Defense Fund (DREDF). (2012, October 10). *DREDF secures historic settlement in National Association of the Deaf, et al. v. Netflix, ensuring 100% closed captions in Netflix's on-demand streaming content within two years*. `https://dredf.org/nad-netflix-reach-historic-agreement/`

European Parliament, & Council of the European Union. (2019, April 17). *Directive (EU) 2019/882 on the accessibility requirements for products and services.* EUR-Lex. `https://eur-lex.europa.eu/eli/dir/2019/882/oj`

European Telecommunications Standards Institute (ETSI). (2021, March). *Accessibility requirements for ICT products and services (EN 301 549 V3.2.1)*. `https://www.etsi.org/deliver/etsi_en/301500_301599/301549/03.02.01_60/en_301549v030201p.pdf`

Halder, S., & Argyropoulos, V. (2019). Concluding chapter: Challenges and strategic pathways to participation, equity, and access. In S. Halder & V. Argyropoulos (Eds.), *Inclusion, equity and access for individuals with disabilities: Insights from educators across the world* (pp. 665–688). Palgrave Macmillan. `https://doi.org/10.1007/978-981-13-5962-0_32`

International Organization for Standardization. (2012, October). *ISO/IEC 40500:2012 – Information technology — W3C Web Content Accessibility Guidelines (WCAG) 2.0*. Retrieved April 26, 2025, from `https://www.iso.org/standard/58625.html`

Jordanoski, Z., & Meyerhoff Nielsen, M.(2023). The challenge of web accessibility: an evaluation of selected government websites and service portals of high, middle and low-income countries. *Proceedings of the 16th International Conference on Theory and Practice of Electronic Governance*, 101–110. `https://doi.org/10.1145/3614321.3614343`

Kanter, A. S. (2003). The globalization of disability rights law. *Syracuse Journal of International Law and Commerce, 30*(2), 241–269. `https://ssrn.com/abstract=2109768`

Open Society Foundations. (2019, December 2). *The story of the first two nations to pass landmark disability rights law*. https://www.opensocietyfoundations.org/events/the-story-of-the-first-two-nations-to-pass-landmark-disability-rights-law

Rogers, M. (2025, April 16). *Government accessibility standards and WCAG 2*. PowerMapper. https://www.powermapper.com/blog/government-accessibility-standards/

Russell, C. (2003). Access to technology for the disabled: The forgotten legacy of innovation? *Information & Communications Technology Law, 12*(3), 237–246. https://doi.org/10.1080/1360083032000198754

Shakespeare, T., Mugeere, A., Nyariki, E., & Simbaya, J. (2019). Success in Africa: People with disabilities share their stories. *African Journal of Disability, 8*, 522. https://doi.org/10.4102/ajod.v8i0.522

Szymanski, C.-F.-X. (2009). The globalization of disability rights law – From the Americans with Disabilities Act to the UN Convention on the Rights of Persons with Disabilities. *Baltic Journal of Law & Politics, 2*(1), 18–34. https://doi.org/10.2478/V10076-009-0002-Z

UCLA WORLD Policy Analysis Center. (2017, January 14 – Last revised: 2024, July 28). *Fundamental rights of persons with disability statistics and charts. Disabled World*. Retrieved April 25, 2025, from www.disabled-world.com/disability/statistics/maps.php

United Nations Department of Economic and Social Affairs (DESA). (n.d.-a). *Convention on the Rights of Persons with Disabilities (CRPD)*. https://social.desa.un.org/issues/disability/crpd/convention-on-the-rights-of-persons-with-disabilities-crpd

United Nations Department of Economic and Social Affairs (DESA). (2024, June). *Disability and development report 2024: Accelerating the realization of the Sustainable Development Goals by, for and with persons with disabilities (Executive summary advanced unedited version)*. https://social.desa.un.org/sites/default/files/publications/2024-06/Final-UN-DDR-2024-Executive%20Summary.pdf

United Nations Department of Economic and Social Affairs (DESA). (n.d.-b). *UNCRPD Article 32 – International cooperation*. United Nations. https://www.un.org/development/desa/disabilities/convention-on-the-rights-of-persons-with-disabilities/article-32-international-cooperation.html

United Nations General Assembly (UNGA). (2006). *Convention on the Rights of Persons with Disabilities*. https://www.ohchr.org/en/instruments-mechanisms/instruments/convention-rights-persons-disabilities

United Nations General Assembly (UNGA). (1994). *Standard Rules on the Equalization of Opportunities for Persons with Disabilities*. https://www.un.org/development/desa/disabilities/resources/general-assembly/standard-rules-on-the-equalization-of-opportunities-for-persons-with-disabilities-ares4896.html

U.S. Department of Justice. (2024, April 8). *Fact sheet: New rule on the accessibility of web content and mobile apps provided by state and local governments*. https://www.ada.gov/resources/2024-03-08-web-rule/

U.S. Congress. (2016, January 7). *Rehabilitation Act of 1973, as amended by the Workforce Innovation and Opportunity Act (WIOA)*. https://assets.section508.gov/assets/files/rehabilitation-act-of-1973-amended-by-wioa.pdf

World Wide Web Consortium (W3C). (2024b, December 12). *Web content accessibility guidelines (WCAG) 2.2*. https://www.w3.org/TR/WCAG22/

Balancing legal compliance and ethical responsibility

Deque Systems. (n.d.). *The Automated Accessibility Coverage Report*. https://www.deque.com/automated-accessibility-testing-coverage/

Law Office of Lainey Feingold. (2025, February 2). *Beware of AI accessibility promises: US federal agency fines an overlay company one million dollars*. https://www.lflegal.com/2025/01/ftc-accessibe-million-dollar-fine/

Forecasting future legal trends in digital accessibility

European Commission. (2021, April 21). Proposal for a regulation laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52021PC0206

G7 Italia 2024. (2024, October). G7 inclusion and disability: The Solfagnano charter. Retrieved May 9, 2025, from https://www.g7italy.it/wp-content/uploads/G7-Inclusion-and-disability_Solfagnano-Charter.def_.pdf

Masterson, V. (2024, February 8). These 6 ‘longevity economy’ principles can help an ageing population live well. *World Economic Forum*. https://www.weforum.org/stories/2024/02/longevity-economy-principles-ageing-population/

United Nations Department of Economic and Social Affairs (DESA). (2022). *World population prospects 2022: Summary of results (UN DESA/POP/2022/TR/NO. 3)*. https://www.un.org/development/desa/pd/sites/www.un.org.development.desa.pd/files/wpp2022_summary_of_results.pdf

White House. (n.d.). Blueprint for an AI bill of rights: Making automated systems work for the American people. *Office of Science and Technology Policy*. Retrieved May 9, 2025, from `https://bidenwhitehouse.archives.gov/ostp/ai-bill-of-rights/`

World Wide Web Consortium (W3C). (2024a, May 16). *W3C accessibility guidelines (WCAG) 3.0*. `https://www.w3.org/WAI/standards-guidelines/wcag/wcag3-intro`

4

User Research and Testing for Inclusive Products

By Julianna Rowsell

Inclusive research is an approach to user research that intentionally includes people from groups who are often excluded, especially people with disabilities, neurodivergent folks, and others facing systemic barriers. It ensures that the products we build reflect a wider range of needs, abilities, and lived realities. Inclusive research isn't just about access; it's about equity. It questions who gets to shape the tools and systems we all rely on, and who gets left out when we don't ask. But inclusion isn't automatic. It requires us to rethink how we design research from the start, who we involve, how we reach them, and what we're willing to change in response to what we learn.

Inclusive research is accessible to different people in different ways, but at its core, it starts with a mindset, one that doesn't default to scarcity, too hard, too slow, or too costly. It sees the richness in reaching and involving diverse participants. When we frame recruitment, the process of inviting people into our research, as a barrier, we limit not just our reach, but our imagination. But when we see recruitment as an act of care, we begin to shift our approach. It becomes a creative and necessary part of equity-centred research, a chance to build trust and relationships, not transactions.

As a neurodivergent researcher, I approach research differently. This can be confusing and feel unfamiliar to those trained in linear methods or traditional UX processes, but this difference is where my insight comes from. I don't just work differently, I sense differently. My brain loops, lingers, and leaps. It notices the edges first: the friction, the reset, the overwhelm. For years, I was told these traits were distractions, something to smooth out, to hide. But they've become my compass. They shape how I hold space, how I listen beneath the words, and how I stay with what doesn't fit.

My methodology isn't linear, because my mind isn't. I move in spirals. I pause. I build in breath. I let silence speak. When things feel unclear, I don't rush to resolve it, I listen. Confusion isn't failure. It's a flare. A signal that something deeper is moving beneath the surface. This isn't inefficiency. It's a slower kind of precision. When research demands clarity too soon, we risk losing what's still unfolding. I've learned to trust the messy, the nonlinear, and the unspoken, because that's where truth tends to live.

Access is where inclusive research begins. It's not just about whether someone can join the study. It's about what it takes to feel safe, to feel seen, to stay. It's relational. It's shaped by power, by history, by context. Before we ask someone to share, we have to ask ourselves whether we've done the work to meet them well. Accessibility, in this light, becomes the scaffolding. Access is the lived experience of it, and includes the following:

- Physical and digital accessibility
- Psychological safety
- Cultural and linguistic respect
- Economic and time-related availability
- Sensory and cognitive ease

What this means in practice is understanding that access is a relationship. It shifts depending on who's in the room, what's being asked, and how power moves through the space. Our work is to shape research spaces where people can show up in ways that feel possible, safe, and meaningful. That doesn't happen by accident. It takes preparation, flexibility, and a willingness to build trust before we ever ask someone to share their insight. When we lead with that, access stops being a checkbox or a feature, it becomes the foundation everything else is built on.

In digital spaces, we've learned that accessibility isn't just about whether something functions; it's about whether it feels usable, respectful, and safe. Research is no different. When practices become extractive, even with good intentions, we replicate the very harms we say we're trying to address. When people are treated as data points, when they're asked to reopen wounds without care or reciprocity, we're not practicing inclusion, we're performing it.

This kind of work isn't fast. It isn't tidy. But it is deeply human. It moves slower, on purpose. It asks different questions:

- Can people participate fully, in ways that honor their needs?
- Do they feel safe, seen, and resourced, not just invited?
- Will their insight shape what comes next or just fill a slide?
- Are we honoring the cost of their contribution or simply collecting it?

These aren't just values, they're working conditions. When we name access, trust, or relational care, we're also naming the responsibility of designing research systems that support them. The sections that follow aren't meant to offer perfect answers, but to shift how we approach inclusive research from the inside out. Because inclusion is not a deliverable; it's a way of being. And before we gather insights or validate what's already been built, we need to ask: Are we making space, or just taking it?

This chapter asks a lot of you, on purpose, because inclusive research isn't just a checklist or a better process; it's a mindset shift. Some parts may feel abstract, messy, or slow. That's by design. But throughout, I've paired reflection with practice, so even the most intangible moments come with tangible tools.

In this chapter, we'll explore the following:

- Recruiting diverse participants for inclusive research
- Adapting research methods for accessibility
- Analyzing qualitative data from diverse user groups
- Translating research insights into inclusive design solutions

These practices don't just make products better, they make the process more just, more human, and more aligned with the future we're trying to build.

Recruiting diverse participants for inclusive research

Let's begin with something simple but not always easy.

If you've been involved in research before, think back to past recruitment efforts. Who showed up? Who didn't? Why might that have been the case? This is an invitation to reflect, not to judge. You should be honest about the patterns we've inherited and the habits we've repeated. Systemic barriers shape exclusion. But they're also upheld by the everyday choices we make, who we think is "reachable," and how we invite, compensate, and respond once people arrive.

Too often, inclusive recruitment is dismissed as too hard, too niche, too slow. But those beliefs don't just limit who we reach; they limit what we believe is possible.

Inclusive research asks us to pause, to pay attention, and to do so with intention. Ask yourself the following:

- Are you supported to move at the speed of trust, even when timelines demand more?
- Where can you build in space for deeper engagement, even within fast-paced realities?

Accessibility doesn't start in the session; it starts long before the first ask. As researchers and designers committed to inclusion, we must adapt our methods. Traditional recruitment often mirrors the same systemic inequities we claim to challenge, filtering out the very people whose insights could shift what we build, how we build, and who it's built for. If we want to do better, we need to rethink how we invite people in, not as an afterthought, but as the beginning of the work. We must adapt our research methods to honor participants' lived realities, preferences, and needs instead of expecting them to conform to traditional research protocols designed for non-disabled norms.

Rethinking recruitment as a relational practice

If we're serious about equity in research, we need to stop treating recruitment like a task to check off and start treating it as a relational practice. At its core, this isn't recruitment. It's relationship-building, rooted in respect, guided by care, and carried by trust. Disabled participants often experience research as extractive, inaccessible, or performative. Shifting to a relational practice is not just best practice; it's essential to earn trust and build equity into how we engage.

As we build our recruitment practices, it helps to keep asking ourselves the following:

- Are our materials accessible?
- Are we compensating fairly for time, energy, and emotional labor?
- Have we built systems of care and safety into the research process?
- Are we building relationships before we ask for insight?
- Are we using trusted community channels or just those easy to reach?
- Have we closed the loop and shown how feedback shaped outcomes?
- Have we created accountability *after* the study ends?

These aren't extras. They're the foundation of a research culture grounded in reciprocity, mutuality, and collaboration, not extraction.

Compensation is also part of how we show respect. When we ask people to share their time, energy, or lived experience, especially when it's shaped by harm or exclusion, we need to recognize what that asks of them. Fair, flexible payment isn't a formality; it's one way we demonstrate that their contributions matter.

Beyond representation: designing from the edges

Representation is about designing with the full complexity of how people move through the world. Especially those pushed off the "happy path," the idealized, simplified user flow that assumes stable access, normative behavior, and little variability.

Participants dismissed as "edge cases," those whose needs or experiences don't fit the assumed norm, often reveal where systems break. Their experiences surface the friction others don't see. Designing from the edges deepens, not narrows, what we build. They're signal bearers. When we take those signals seriously, we're not narrowing the experience; we're deepening it.

A neurodivergent participant might not stumble because a flow is unclear, but because it demands energy they don't have. It's not just layout, it's overload, sensory input, and pattern complexity. When we listen attentively, we begin to understand not just what's breaking but why.

Research tends to over-index those who are easiest to reach: folks with stable internet, high digital literacy, and comfort with research norms. But that filters out the very people whose perspectives could transform how we build. Inclusive research asks us to hold complexity, not flatten it, even more so when studying disability and access needs. Too often, disability research is treated as supplemental, anecdotal, or reduced to statistical slices in broader studies. This misses the nuance of lived experience and erases variation, pattern, and cultural context.

We must build systems that reflect real, shifting, and layered lives, with outcomes flexible enough to hold people's realities, which are nonlinear, dynamic, and deeply human.

There's no universal checklist for inclusive recruitment and no step-by-step formula that guarantees care, reciprocity, or trust. But there are questions we can keep asking, patterns we can notice, and practices we can return to. Think of this not as a prescription but as a starting point – something to revisit, reshape, and build upon in your context. Consider these questions:

- Have I involved community members in shaping the research process?
- Are my recruitment materials accessible (language, format, delivery)?
- Have I used trusted, culturally relevant channels to share opportunities?
- Does my compensation reflect participants' time, effort, and lived experience?
- Have I offered multiple ways to participate (synchronous/asynchronous, verbal/non-verbal)?
- Have I closed the loop by sharing outcomes or updates with participants?
- Am I respecting access needs that intersect with race, class, gender, language, or geography?
- Have I built in flexibility around timing, energy, and participation boundaries?
- Have I made space for pauses, emotions, or nonlinear expression in sessions?
- Are participants shaping what questions are asked, not just answering them?
- If this process were reversed, would I feel safe, respected, and valued?

What matters is applying them thoughtfully, in context, and with the flexibility to adapt. When we design *with* those most impacted, not just for them, we create systems that are more just, more adaptive, and more alive. This isn't about surface-level fixes. It's about refusing to default. It's about meeting people where they are, holding what they carry, and letting it shape how we build. It's about honoring the humanity behind the data. The most transformative insights don't live in the average. They live at the edge.

And sure, you won't always get it right. But inclusion isn't a checklist, it's a practice. What matters is that we keep choosing it, again and again.

Centering disabled lives through reciprocal recruitment

Recruiting disabled participants isn't just about inclusion, it's about shifting the terms of engagement. When we center disabled lives in recruitment, we begin to challenge the unspoken assumptions that exist in how, when, and where we bring people in. Recruitment is often transactional: time in exchange for a gift card or insights in exchange for a deliverable. But when research stops there, it risks becoming exploitative. That leaves people feeling used, not valued.

Reciprocal recruitment asks something more of us. It asks us to slow down, to build trust before the ask, and to create spaces where people can show up fully, not just to answer questions, but to shape them. It means inviting people in not as subjects, but as collaborators. It also means recognizing that knowledge doesn't only live in transcripts. It also lives in a stillness, a breath, a drawing, a moment of mutual understanding. For example, a team working with a Deaf community co-designed a research plan alongside a local Deaf advocacy group, hired interpreters chosen by participants, and allowed flexible participation over multiple sessions. The result wasn't just better data, it was a model for respectful, sustainable engagement.

While recruitment always involves some exchange, it doesn't have to be extractive. When it's grounded in trust and reciprocity, it shifts from a one-way transaction to a shared investment. It's relational. And it shifts our goal from participation to partnership, and from gathering insights to building something that gives back.

To support this shift, here are a few foundational principles that guide inclusive recruitment practices:

- Build relationships before recruiting. Start with trust, not outreach.
- Compensate fairly and flexibly. Honor time, energy, and emotional labor in ways that reflect people's realities.
- Respect intersectionality. Understand how overlapping factors, such as disability, race, gender, class, and language, shape people's experiences and influence what they need to feel safe and supported. For example, a white wheelchair user navigating an urban center will likely face very different barriers than a racialized autistic person living in a rural area. Intersectionality reminds us that access is never one-size-fits-all; it's influenced by overlapping systems of power, discrimination, and privilege. Inclusive research must account for these layered realities or risk reinforcing the very gaps we seek to close.

- Co-design the process, not just the output. Share power from the beginning.
- Avoid extractive dynamics. This means going beyond scripted questions and one-off engagements. When we treat participants as data sources instead of people, we risk commodifying their lived experiences.

Instead of expecting folks to come to us, go to them. Listen, return, and show up with integrity, not just once, but over time. We can't co-create from a distance. That's why some of the most grounded, respectful research happens through partnerships with communities that already hold trust, advocacy groups, cultural organizations, and grassroots networks.

One powerful example is, in one of my research engagements with a community of disabled and neurodivergent participants, I partnered with a grassroots advocacy group to co-design the recruitment process from the start. Rather than using formal consent forms or standard screener scripts, we developed plain language materials together. Participants helped identify not just accessibility needs but how they wanted to be engaged, with options such as voice notes, drawings, asynchronous participation, or sensory-friendly formats. We compensated people based on their preferred method and energy levels, with no penalty for opting out midway. Interpreters and support people were invited at the participants' request, not ours. The result wasn't just more honest insight – it was a deeper relationship grounded in trust, where participants could influence the research itself, not just respond to it.

This approach demonstrates the following:

- Careful and collaborative planning
- Flexibility and agency
- Rejection of one-size-fits-all engagement
- Mutual respect and access as the starting point, not as an add-on

This kind of reciprocity isn't theoretical; it's being practiced in real-world projects that share power with disabled participants and value their ways of knowing.

This example isn't just a moment I remember, but moments that reshaped how I work. I share them here not to prescribe but to help others recognize these moments in their own lives. The work of inclusive research lives in these shifts – in how we listen, what we notice, and how we carry it forward.

In the next section, we'll explore how to adapt research methods for accessibility, shaping not just who we reach but how we listen, engage, and build research environments that honor complexity, support participation, and move at the speed of trust.

Adapting research methods for accessibility

Adapting research methods isn't just a technical challenge; it's deeply human. If we're serious about centering disabled lives, we can't just set up a session and hope for the best. We need to do the work before the ask, to make sure our methods, tools, timelines, and structures are usable, flexible, and safe.

Participation is never neutral. For many disabled participants, showing up is an act of vulnerability, shaped by a history of being left out, misunderstood, or expected to adapt without support. There's a long legacy of research that extracts, simplifies, or forgets. Experiences shared generously have too often been met with silence.

I remember a time in my career when I worked on a team made up mostly of disabled folks. Three of my colleagues were blind or had low vision, and each navigated the digital world in completely different ways:

- One used high-contrast mode and zoomed in 800% on everything, relying on screen magnification and visual cues, not a screen reader.
- Another had been blind since birth and used a screen reader alongside a braille display, processing content linearly, precisely, with little need for spatial orientation.
- The third had lost their sight in their 30s and used a screen reader too, but still anchored their understanding of digital layouts in visual metaphors where things were *supposed* to be on the screen, based on muscle memory and residual sight.

This wasn't a theoretical insight. I saw it firsthand: three blind colleagues, one team, and three entirely different ways of navigating the same space – the same disability label, but different timelines, tools, and truths.

It changed how I understood accessibility. It made clear that there is no "universal user" and no one right way to engage, and it reminded me that adaptation is not optional, it's foundational. This is why we can't rush research, why "best practices" aren't always best, and why trust isn't built in the sign-off but in the small decisions: how we show up, how we ask, how we hold space.

That sense of being valued deepens when participants are treated as collaborators, not subjects or data points – when their stories shape the work and don't disappear into a folder. And it holds – really holds – when we follow through, and when people can trace the thread between what they offered and what changed because of it.

That's why accessibility can't be an afterthought. It has to be built in from the beginning – not just in tools or timelines, but in how we listen, how we follow through, and how we honor what's shared.

Creating safer and more empowered research spaces

Psychological safety isn't a bonus, it's the foundation of ethical research. When we invite people to share lived experiences, especially ones shaped by exclusion, harm, or marginalization, we're asking for something vulnerable. That kind of sharing can't happen without mutual respect.

Creating that trust doesn't happen by accident. It requires intention. It means making sure people feel heard, respected, and safe enough to express their truths without fear of judgment or dismissal. Not just because it's good practice, but because it's the only way to honor what's being offered.

This looks like offering content warnings when we know a topic might bring up painful memories. It looks like giving participants full permission to skip questions, take breaks, or step away entirely. It looks like naming the emotional labor involved and responding to it with openness.

Following up matters. Sometimes, that means sharing resources, such as a list of local mental health services, disability peer groups, or follow-up materials in plain language. Sometimes, it means offering support in the form of a check-in message, responding to participant reflections, or ensuring there's a trusted point of contact beyond the session.

We, as researchers, have to let go of urgency, or at least question which kinds of urgency are self-imposed and which are truly immovable. We may not be able to slow every deadline, but we can choose to center care in the decisions we do control and the narratives we seek to share.

Moving at the speed of need means adapting to people's lived rhythms. It means knowing that energy fluctuates – that pain, fatigue, caregiving, brain fog, or burnout don't follow our research calendars. It's not just about offering flexible schedules or adding alt text to a slide deck. It's about recognizing that access isn't fixed. It shifts, and so must we.

Take something as simple as a text-based wireframe. For a dashboard, this might look like a linear narrative describing what's on screen: "At the top, a search bar. Below that, two side-by-side panels: the left shows recent files, the right shows project collaborators." No visuals. No layers. Just a map made from words. It's a way to make space for those who process differently, who use screen readers, or who just need more clarity. This won't always be the case, and words can get complex, but this is a path that can help in the right context. It can also serve as a method of including people with disabilities in research at lo-fi or prototyping stages.

That's what moving at the speed of trust looks like. Not flashy. Not fast. But deeply intentional. When people feel expected, not exceptional, access isn't something we offer. It's something we build in from the beginning. Trust isn't built in a single session. It grows in the quiet details: in the format of the invitation, the pace of the process, and the tools that say, *"We thought about you before you arrived; you weren't just a data point here."*

Embracing lived experience

Accessibility in research isn't just about making sure a tool works or a session runs on time. It's about noticing, tuning into what's emerging in the moment, and responding with care. It's about offering options, not ultimatums – designing in ways that honor lived realities, rather than expecting people to perform around them or shrink themselves to fit the method.

True accessibility demands more than compliance. It nudges us into a more relational practice – one that adapts to difference, welcomes friction, and remains open to change. Research that shifts to meet the body, mind, rhythm, and knowledge of each participant becomes more than just a method – it becomes a form of trust-building, and that's where the real insight lives.

Usability testing should reflect how people actually live, not just how we imagine they do. That means inviting participants to use their own devices and their own assistive technologies in the spaces they already move through each day, not in unfamiliar labs or sterile setups that strip away context, comfort, and the small details that shape how access really works.

Real clarity emerges when people are grounded in their tools, their environments, and their own rhythms, and when they're not contorting themselves to fit a process that wasn't built for them. Contextual inquiry, observing how someone navigates a product in their everyday life, reveals what a controlled setting can't. The lighting. The background noise. The interruptions. These aren't distractions. They're part of the story. And for folks with sensory sensitivities or attention differences, they're often the deciding factor between *usable* and *unusable*.

That's why inclusive workshops are designed with flexibility in mind. They make room for multiple modes of expression, whether that's typing, speaking, sketching, recording, or joining asynchronously. They build in breaks. They welcome people to enter at their own pace. They offer softness, not strain.

Remember that co-creation should never come at the cost of someone's body, capacity, or well-being. Too often, workshops ask people to be "on" for hours at a time, assuming stamina, quietness, structure, and constant focus. But that model leaves so many people out. And it doesn't have to be that way.

When we remove the barriers to participation, we don't just create more space, we create the conditions for people to feel seen. And when people feel seen and supported, they show up more fully.

It asks deeper questions:

- What happens when someone's internet is unstable, or their timezone makes "standard hours" inaccessible?
- How do sensory sensitivities impact someone's ability to participate?
- What would it take to create an environment where people feel psychologically safe sharing their truths, not just once, but throughout the process?

When we design for flexibility, we shift the weight of access. We stop asking participants to stretch themselves to meet rigid systems. Instead, we take responsibility as researchers, facilitators, and designers to move toward them.

That's the difference between offering access and practicing equity. One keeps the door unlocked. The other remakes the room.

Inclusive research in practice, even when resources are tight

Not every team has the resources, funding, time, or headcount to implement every inclusive research recommendation at once. But inclusive research isn't all or nothing; it's built through consistent, intentional choices. Inclusive research doesn't have to wait for permission, perfect timing, or the next fiscal cycle. There are small ways to start where you are, with what you have, while staying true to your values. Let the work be imperfect and in motion. Here are a few small shifts that can move your practice forward, even in constrained environments:

First of all, start small:

- Use plain language in recruitment materials. It costs nothing and boosts accessibility.
- Offer asynchronous participation (e.g., email, voice memos) – great for neurodivergent or time-constrained participants.
- Pilot with 1-2 participants from an underinvested group. Not ideal, but better than complete absence.

- Create a "trust prompt" in your screener or intro email. Ask: "What do you need to feel comfortable participating?"
- Budget $50 for mutual aid. Buy someone a transit pass, a meal, or a sitter to make participation possible.
- Share back findings. A short thank-you email and a bullet list of what you learned closes the loop and builds trust.
- Adapt consent as a conversation. Even a 30-second verbal check-in before you start can feel different than a formal waiver.

Plus, work with what you have:

- Reallocate 5-10% of your existing time. Use it to improve onboarding, feedback loops, or protocol accessibility.
- Look at existing tools. Can you build inclusion into the tools or processes already in use (e.g., Miro boards with alt text)?
- Invite a community partner to review your plan. One quick async pass from a trusted voice can reframe assumptions.

When you can't do everything, you have to ask yourself what is the minimum change you can make to reduce exclusion in the context that you are in. Even a single choice can build cumulative trust. You don't need perfect conditions to move in the right direction.

Providing flexible communication methods

Not everyone can show up on cue. People living with chronic illness, neurodivergence, disability, or just the weight of life, don't always move in straight lines. Energy comes and goes. Focus fades and returns. Pain flares. Caregiving responsibilities shift. There's no tidy schedule for that.

And yet, research sessions are often built like little performances: tight time blocks, fixed formats, unspoken pressure to stay on, stay coherent, and stay useful. Yes, time, capacity, and budget are real constraints. But so are the costs of exclusion, poor data, misaligned solutions, or entire communities left out. More inclusive practices don't always cost more, sometimes they just require us to notice what's possible.

An inclusive approach starts by letting go of that rigidity. Let people reschedule, without guilt. Break longer sessions into smaller pieces. Make it okay to slow down and come back later, to choose rest over pushing through. Also, remember that people don't always engage in the ways we expect, and that's not a problem. That's the point.

For some, showing up means writing instead of speaking, especially if talking on demand is hard, or if language itself feels like a barrier. Others might need more time, not because they're checked out, but because their energy comes in waves. Fatigue, pain, burnout, and brain fog – none of these follow a research calendar. Showing up isn't linear.

This is why inclusive research can't offer just one path in. It needs to offer options. Real ones. That might mean the following:

- Sending a few written prompts in advance instead of jumping straight into conversation
- Inviting voice memos from folks who prefer speaking over typing
- Using asynchronous tools such as shared docs or messaging threads
- Offering longer timeframes for people who need pacing that matches their lives
- Letting people opt for phone over video, especially if tech or energy is limited
- Using 1:1 conversations instead of group calls if that helps reduce overwhelm
- Keeping instructions or prototypes in plain text, not just images or slides
- Leaving room for processing, pauses, or breaks during live sessions

These aren't "access features." They're basic signs of respect. When we stop forcing everyone through the same narrow door, we start hearing more honest reflections – ones that don't have to push through discomfort just to get to the surface. Access isn't just about whether someone can be there, it's about whether they're welcomed in a way that doesn't cost them more than they can give.

Closing the loop with participants

Building feedback loops is essential for maintaining accountability in inclusive research. Participants need to see how their experiences influence product development to trust the research process.

Where it's legally or practically possible, regular post-research updates and ongoing summaries help close the loop. Even when full product updates can't be shared, a simple follow-up email summarizing what participants contributed and what the team is learning can still demonstrate that their voices matter.

Inclusive research is not just about who is invited to participate; it is about ensuring that their contributions lead to real change, reinforcing relational integrity, and fostering long-term relationships between researchers and research participants.

Inclusive recruitment isn't just about who we invite; it's about how we engage participants throughout the process. Disabled participants and marginalized communities often carry emotional labor and vulnerability into research spaces. They share personal experiences of exclusion and harm, and it's our responsibility to ensure they leave the research feeling seen, heard, and valued.

Here's how to embed systems of care and safety into your research practices:

- Provide emotional check-ins before, during, and after sessions
- Offer resources for mental health support or community care
- Ensure accessible communication channels for ongoing trust-building

Ensure participants can follow up after the session to ask questions, provide feedback, or voice concerns. Providing follow-up emails with summaries of research findings and next steps helps participants see how their contributions are making an impact. Disabled participants often relive exclusionary experiences when participating in research, which can be emotionally taxing.

Research as a dynamic and human-centered practice

As you can see, inclusive research isn't just about improving our methods; it's about rethinking our ways of outreach. It's about moving from extraction to collaboration. From asking for input to building something *with it*. The goal isn't to collect stories and file them away. It's to make sure those stories lead to something real – something that matters.

Co-designed testing means involving participants not just in evaluating a product, but in shaping the research approach itself, such as deciding what questions to explore, what accessibility needs to plan for, or what success looks like from their perspective.

When we embed accessibility into the research process through flexible participation, co-designed testing, trauma-informed practice, and emotionally safe spaces, we're not just adding rigor. We're making the work more human. We're removing the friction that people are so often expected to push through just to be heard.

But that's not the finish line.

Once the insights come in, we have to carry that same energy, that same accountability, into how we *make meaning* of what we've heard. Inclusive research doesn't end with participation. It lives in how we interpret, how we hold complexity, and how we notice patterns without overly simplifying them.

And it lives in how we push back against the dominant narratives that have long shaped what's seen as "valid," "valuable," or "objective" in the first place. Research that begins with care should end with accountability. That's where meaning lives and where change can take root.

In the next section, we'll look at what it means to analyze qualitative data in ways that honor the full context, center lived experience, and open up space for a more honest, more equitable understanding of what really emerges when we listen well.

Analyzing qualitative data from diverse user groups

Data is not neutral. Every piece of research data carries with it the weight of its context, the biases of its collection methods, and the lived realities of those who provide it. We must extend beyond the surface to the human story in the data. What we include in our datasets or code directly influences the conclusions we draw and the actions we take. When we analyze qualitative data from diverse user groups, we must approach this with humility, understanding, and awareness of the potential power dynamics.

When analyzing data from marginalized communities, understand how your own biases may influence interpretation. To help counterbalance this, we need to prioritize participant voices over our assumptions or beliefs:

- Sometimes we ask about identity directly (race, disability, gender, or income) because it's part of our recruitment goals or ethical obligations.
- Other times, participants choose what to share, and we don't always have the full picture.
- Both realities shape how we analyze. When identities are named, we gain more context for interpretation. When they're not, we must be careful not to default to a single-issue lens.
- Lived experience is layered. A quote about screen reader frustration might also carry the weight of economic precarity, cultural mistrust, or systemic neglect.
- Intersectionality doesn't mean we have to decode everything; it means we stay aware that people don't show up with just one story. For example, a person with a vision-related disability may also be a single parent or part of the 2SLGBTQ+ community (the use of "2S" acknowledges Indigenous understandings of gender and sexuality, which are often erased in mainstream frameworks). All of which shape their experiences and how they navigate the world.
- Reflexivity in research means holding space for the complexity we *see*, and the complexity we *don't*.

When we collect data for underinvested communities, we are not just gathering information; we are gathering collective experiences that have been shaped by their complex identities. These intersecting identities shape their experiences in ways that can't be neatly categorized into one demographic box. Yet, too often, research outputs reduce these experiences to snapshots by pressing them into a neat shape without nuance.

Neurodivergence as method: how lived experience rewrites the rules

I used to think research had to look a certain way: poised, prepared, and perfectly structured. But that version of research often left my body behind. As someone with ADHD, my brain doesn't move linearly. It operates a bit like a constellation, zooming in and out at will. It hyperfocuses on patterns others miss and then collapses under too much sensory input. That used to feel like a liability. Now, it's the compass I trust most.

It's why I build in long pauses when I facilitate. It's why I offer asynchronous options. I send questions in advance, not just for participants, but because I know how much it helps *me* to engage on my own terms. I've unlearned the idea that good research is fast, tidy, or neutral. The research that lives with me the longest, the kind that stays in my bones, is the kind that made space for slowness, for overwhelm, for disorientation, and still kept listening.

There's a moment after research where the pace tends to pick up. Synthesis. Reporting. Action. But if we rush here, we miss what actually matters: how we hold what people gave us.

This part, the interpretation, isn't just about themes or codes. It's about meaning. And meaning isn't clean. It's multifaceted. It contradicts itself. It comes with disruption, with hesitation, and with emotional weight that doesn't always show up in transcripts.

Thematic analysis, in many research spaces, becomes a process of compressing, of tidying up what was messy so that it "makes sense" to someone else. But in inclusive research, we can't afford to smooth everything down. We need to leave room for contradiction, for multiple truths, for the stuff that doesn't fit easily into a sticky note on a wall.

Especially when we're listening to disabled folks or people navigating layered identities, what they've shared didn't come cheap. It's shaped by years of friction, adaptation, and being misunderstood. So when we analyze that data, we have to ask: Are we making meaning, or are we erasing it?

We know the trade-offs: limited time, tight budgets, institutional barriers. Not every project allows for sweeping change. But every project allows something. And those somethings add up.

The responsibility of meaning-making

It's not enough to witness what others offer up, we have to consider how we interpret that. Too often, insights get flattened in analysis, reduced to pull quotes that strip context, emotion, and nuance. But inclusive research asks more of us. It asks: How are we holding meaning? What are we doing with the truths people trusted us to carry? This kind of interpretation can't be rushed or neutral. It has to be relational, reflexive, and deeply aware of power.

So, I return to a few core questions that guide how I work:

Are we interpreting in a way that reflects the weight of what was shared?

One time, a participant shared that they had stopped using a key feature of our product because it wasn't compatible with their screen reader. The previous iterations had been accessible to them, and they felt they had lost something that gave them agency and autonomy. They described feeling embarrassed and exhausted from constantly having to find workarounds. They spoke plainly – almost like it wasn't worth mentioning and like they expected nothing to change. But we knew that feeling: the disappointment that comes when yet another feature slips out of reach for someone who once relied on it. Instead of logging this as a bug or "edge case," I worked with the team to recognize it as a signal of exclusion. We reframed the insight as a design principle: "If one person can't use it, no one fully can." This wasn't just a technical fix; it was a reminder that accessibility is about respect, not compliance. And it shaped how we prioritized changes across the product.

I've carried stories like this as memory in my body – not just as notes or tags in a research repo, but in the quiet heaviness after someone said, "I've never been asked that before." That kind of honesty doesn't belong on a slide. It deserves reverence. It deserves to be held with the kind of care that says: Your truth is not just valid, it's the lens we should be building from.

For me, interpreting data has never been about sorting quotes into neat buckets. It's about listening for what's pulsing underneath. What they couldn't quite say. What they trusted me to feel anyway. And it means asking myself, not just the data, what is this trying to teach us if we're brave enough to stay with it?

Have we held the emotional, cultural, and systemic context, not just the quote?

As a neurodivergent researcher, I've shown up to sessions while masking, flaring, looping, and doing what it takes to hold space even when my own system is overstretched. And I've interviewed people who've done the same. Who answered with precision, but whose voices carried fatigue. Whose hesitations held more than language. I've learned to pay attention to those layers: the cultural context that shapes how someone speaks, the systemic weight they've carried into the room, the way a single story often stands in for so many others. We don't need more quotes that sound good. We need an interpretation that honors the conditions they emerged from and refuses to erase them in the process.

In one session, a participant had been engaged and open until I asked a question about their past experiences with support services. They went quiet. Not abruptly, just… inward. Eyes down, the tone shifted, a short answer followed, then we moved on. I didn't flag it as "participant disengaged" or drop the data point. I made a note: *"Tone changed here, revisit."*

Later, in synthesis, that moment helped shape a broader insight into emotional safety in research settings, especially when asking about institutions people may have been harmed by. It reminded me to build trauma-aware prompts, to wait before probing, and to read data not just by what was said, but when something stopped.

That experience changed how I write protocols. In later studies, especially with disabled and neurodivergent participants, I began including explicit pauses between questions, and added optional "skip" buttons or prompts such as *"Is this something you're comfortable talking about today?"*

I also started debriefing *how the research felt*, not just what was shared. That surfaced patterns: which topics led to withdrawal, which formats felt safer, and when people opened up more in asynchronous follow-ups than in live sessions.

These adjustments weren't just for participant comfort; they led to richer, more honest data. The small design changes, offering choice, naming emotional labor, and holding back from immediate follow-ups, helped shift power back to the person sharing. What I learned is that silence is feedback. And when we listen to it, it teaches us how to design research that earns, rather than demands, people's trust.

Have we left room for stories that don't align with our assumptions?

Sometimes, people leave questions blank. Not because they don't care, but because they do. Because it's too close, too much, or too tiring to explain again. I've come to see that as data, too. Ambiguity isn't failure. The disclosures that contradict or resist tidy arcs are often the ones I trust most. They remind me: this isn't about clarity, it's about complexity. And if our systems can't hold that, maybe it's the systems that need changing.

Responsible interpretation isn't passive. It's a form of authorship. It shapes what gets built, believed, and backed. We don't just honor the stories, we participate in reshaping the conditions that made them hard to tell in the first place.

If you're wondering what this looks like in practice, that's the point. This work asks us to stay with what doesn't have easy answers. But that doesn't mean it can't be actionable. Later in this chapter, I offer tangible methods, recruitment approaches, accessibility prompts, and synthesis strategies that help us hold complexity without collapsing it.

As we analyze qualitative data, we need to pay attention not just to what was said, but how it was shared. Did someone hesitate? Skip a question? Go silent? These are not gaps, they're signals. Tag them. Follow up. Use them to resist sanding down the complexity that often surrounds disability, trauma, or exclusion.

Sometimes, insight doesn't land with clarity. It lands with weight. Our job isn't to extract it, it's to hold it. To carry what someone entrusted us with, even if they didn't say it out loud. When that happens, don't rush to define or resolve it. Mark it. Return to it. Let it linger long enough to reveal what it's asking of you. Not everything speaks in clean lines or tidy quotes. Some truths arrive quietly, asking us to stay with them a little longer before we decide what they mean.

Dynamic data in a fluid world

Inclusion isn't just about who's at the table; it's about how meaning is made, and whose truths are allowed to shape the system. In research, the way we hold data is never neutral. What we prioritize, discard, aggregate, or interpret reflects not just what we've learned, but who we've centered and who we've overlooked.

As Payal Arora, a digital anthropologist focused on the Global South, reminds us, global digital narratives often ignore the informal, the adaptive, and the messy lived realities of the majority world. Jutta Treviranus, a pioneer in inclusive design, urges us to see inclusion not as exception-handling, but as the root of meaningful innovation. Both challenge the tyranny of the average and call us to treat difference as a starting point, not an afterthought.

Still, research often falls into familiar analytic traps:

- **Insight isn't static**: The trap here is time. One session becomes the story. A single moment is mistaken for a fixed truth. But needs are not static, especially for disabled, neurodivergent, or systemically excluded participants. They're shaped by shifting energy, evolving tools, and changing contexts. When we freeze what we've learned in place, we design for what was, not what's emerging.
- **From edge cases to systemic signals**: The trap here is the discard. Outliers get framed as noise, deviations from the "norm," when they're often the clearest signals of where systems are breaking. What seems like friction for a few is often a flare for deeper design flaws.
- **When insight becomes harm**: The trap here is overreach. Group-level trends get applied to individuals. Nuance collapses into generalization. Statistical discrimination turns differences into stereotypes. This is how systems quietly replicate harm under the guise of objectivity.
- **Over-relying on dominant trends**: The trap here is overrepresentation. What's statistically common often reflects the most reachable participants, not the most excluded. When we treat the average as truth, we marginalize the very perspectives and lived knowledge that could drive meaningful change.
- **Flattening nuance into categories**: The trap here is forced coherence. Thematic analysis can tidy up contradiction and tension, but lived experience, especially at the margins, rarely fits cleanly. Instead of forcing coherence, we can let complexity stand, treating silence as presence. Pauses, hesitations, skipped questions – these are often signals, not gaps. They might hold fatigue, mistrust, or truths that don't fit our frameworks. Listening well means listening between the lines.

These aren't technical oversights – they're decisions about what we count as knowledge. These are methodological and interpretive choices that shape what's valued, validated, or dismissed in the process. They reveal what we allow to count as valid knowledge, and what we're still willing to overlook. To move through them, we need to shift how we hold what's shared – not as something to extract, reduce, or resolve, but as something to stay in a relationship with. Meaning lives in friction, contradiction, and refusal to be boxed in.

Inclusive analysis refuses to resolve complexity too quickly. It lingers. It asks better questions.

It makes space for multiple truths. To move through these analytic traps, we need more than awareness; we need methods that hold contradiction, tools that don't force consensus, and patterns of practice that slow us down when our instinct is to simplify. Some ways to do this include the following:

- Revisiting findings over time, especially with participants from historically excluded groups
- Creating feedback loops that include participant check-ins and community sense-making
- Designing analysis frameworks that leave room for anomalies rather than collapsing them
- Making silence visible in datasets, not as a gap, but as a meaningful flag

Understanding alone doesn't create change; it's what we do with it, how we push it forward, translate it, and let it reshape what gets built that determines whether our research stops at understanding or moves toward impact.

Next, we'll look at how to carry insights forward into design, not as inspiration, but as a force for transformation, so that what we've heard shapes what gets built.

Translating research insights into inclusive design solutions

Translation is where things often fall apart. Research gets done, what's learned gets shared, but nothing changes. Reports sit in folders. Findings get watered down into vague "takeaways." Product teams keep moving. And the real-world impact, the kind that shifts experience, never happens.

To move from discovery to action, we have to take on the role of translator – not just handing things off but bridging the space between what was heard and what needs to be built. That's not a handoff; that's reciprocity in practice.

Too often, insights get treated like static observations instead of something that should reshape the work. They get lost in speed, in roadmap pressures, in the "we'll get to it later" pile. But the point of inclusive research isn't just to name barriers. It's to help remove them.

This isn't just about fixes. It's about rethinking the patterns, defaults, and assumptions we've let sit too long. What's surfaced only matters if we carry it forward. Otherwise, it's just noise.

Moving from fixes to futures

There's a tension between short-term fixes and long-term futures. Product teams often feel pressure to move fast, to patch what's broken, creating confusing flows and inaccessible features with high-friction interfaces. The real value of research lies in what it surfaces underneath: the systems, patterns, and assumptions that need to shift.

Take, for example, research with neurodivergent users – autistic, ADHD, dyslexic, or otherwise. The friction often isn't just about complex navigation. It's about cognitive load, energy management, and how our brains process structure, sequence, and flow. A quick fix might be fewer clicks or simpler labels. That might help. But it doesn't get to the root.

A deeper response asks us to step back, to rethink the entire architecture:

- Are we organizing content in a way that's intuitive, flexible, and responsive to diverse ways of processing?
- Are we honoring the ways people scan, pause, hyperfocus, circle back, or shut down?
- Are we still designing for one dominant way of thinking?

When we design for neurodivergence, we don't narrow the experience; we expand it. We ease friction. We open possibilities. Neurodivergent feedback often points to deeper design patterns – how content is sequenced, how options are presented, and how choice and overwhelm are managed. That's not just inclusive. That's good design.

But designing for the future means letting go of old tools that distort who people are. Personas, those neat little profiles, often feel helpful. But they collapse complexity. They shrink lived experience into static traits, turning vibrant realities into stereotypes.

Instead of asking *"Who is this person?"* we might ask *"How are they approaching this moment?"* Are they overwhelmed? In deep focus? Moving slowly or impulsively? Archetypes and mindsets offer more flexibility. They don't define people, but they give us ways to recognize patterns of being, such as Navigators, Improvisers, and Scanners, without locking anyone into a single identity.

When we move away from fixed profiles and toward layered, contextual insight, we begin designing for *ways of being*, not just static descriptions. That's where inclusive design gets its power, not from knowing someone perfectly, but from making room for how they change.

The stories shared so far aren't side notes; they're the work. They show what it looks like to stay with discomfort, to read beyond what's said, and to let insight reshape the systems we operate in. If we only collect data but don't let it change us, we're not practicing inclusion, we're performing it. Designing for both existing and future users requires flexibility. Designing for now *and* what's next.

Insights as relationships, not output

Creating feedback loops where research findings are shared with participants builds continuity and long-term engagement. We might experience this when a product designed with blind users in mind might technically pass a WCAG compliance check, but if real users struggle with its structure or interaction model, it's still inaccessible.

When someone tells us where a system failed them, that isn't feedback, it's a beacon. It's a signal that says, *"This didn't work for me, and I'm still here, trying to help you do better."* What we do next matters.

Without care, even inclusive products can replicate harm. I've seen tools pass every compliance check, yet still feel unusable. I've watched teams celebrate optionality, five ways to do a task, but miss the point entirely because none of them truly meet people where they are. That's not flexibility. That's noise. It's the difference between adding features and building alignment.

But that kind of connection doesn't happen by chance. It requires a system that carries insight across research, design, and implementation, without losing its shape or soul. That's where I bring in a framework I've used across both public and private sector work. I call it *roots, routes, refuge, and realities*:

- **Roots**: Keep us grounded in community, in history, in where this work comes from, and who it's meant to serve
- **Routes**: Map how insight travels through sprint planning, design audits, and decision logs so it doesn't disappear in translation
- **Refuge**: Creates space for care, so that researchers, participants, and designers aren't stretched to breaking
- **Realities**: Anchor us in what's true, not what's convenient, or clean, but what's actually lived

I've used this model to compost insight – not to discard it, but to let it break down, be absorbed, and feed the next season of work. This happens when teams risk treating inclusive insight as transactional, or when the most important feedback is too inconvenient to act on quickly. This isn't a fix-all, but it's a rhythm.

Remember, when someone shares how a system failed them, treat it as a turning point not just a data point. How you hold that feedback will shape whether they trust you, or anyone else, with it again.

Designing from disability, not for it

Inclusive design doesn't begin with retrofitting fixes onto broken systems; it starts by centering disabled people from the outset. Voice input, for example, was originally built for accessibility, and now powers mainstream tools such as Siri and Alexa. Designing from disability often leads to innovations that benefit everyone.

One approach I've found useful is building archetypes and mindsets rooted in lived experience – tools that reflect how people actually navigate complexity, not just how we imagine they do. These frameworks feed design patterns, usability heuristics, and decisions that respond to intersectional needs.

Insight is only powerful if it travels. What we've heard must shape decisions, influence investment, and challenge the defaults that have long excluded disabled users.

Typically, research identifies the barriers, design starts to build solutions, and the product moves forward with delivery, but those threads don't always weave into something whole. Misalignment happens. And when it does, inclusion becomes performative instead of transformative.

To bridge that gap, we need to work with new approaches – all of us. Researchers need to structure observations in ways that are actionable and visible across teams. Designers need to bring accessibility upstream, embedding it in systems, not just screens. Product teams need to make space in their processes, in their priorities, for accessibility to be part of the core build, not the backlog.

That means doing the following:

- Integrate accessibility into design systems, pattern libraries, and components from the start, so it's not a retrofit every time something new ships
- Ensure research insights inform product roadmaps, success metrics, and sprint planning

But this isn't work we do alone. Inclusive research is a collective practice – intentional, ongoing, and relational. It requires collaboration across teams: researchers, designers, engineers, product leads, and the communities we're designing with.

Supporting designers with actionable findings

Designers work where research meets execution. To support them, we need tools that translate intention into tangible action. Many receive research themes and patterns too late in the process, making it difficult to implement foundational accessibility improvements. Others lack clear accessibility and usability guidelines that translate research into tangible design decisions. And even when they build inclusively, they rarely get validation that their designs work in real-world use cases with diverse users.

To better support designers in translating research into action, product teams can equip them with practical, evidence-based tools that bridge intention and execution:

- Inclusive design guidelines tailored to a product's needs
- Design system recommendations that translate participant learnings into reusable patterns
- Experience journey maps that reflect real user behaviors, not just demographics
- Accessibility user stories and acceptance criteria to stress-test designs before they ship

For example, it's not enough to say, "Users with mobility impairments struggle with small touch targets." That may be true, but it doesn't give teams what they need to design differently. Instead, researchers can make it real by recommending minimum tappable sizes, sharing annotated mockups that show accessible interactions in context, or offering prototypes tested with real tools, switch controls, voice navigation, and adaptive inputs.

This kind of specificity empowers. It invites the team into the solution. It helps designers move from awareness to action, giving them the clarity and confidence to build in ways that extend better utility to more people. Designers are among our closest collaborators in this process. Research doesn't just inform design, it helps to reshape the underlying defaults of the product system. Inclusive intent is operationalized in how we work together. It's how we move from asking for better to building it, and is how inclusion takes root.

Summary

This chapter followed inclusive research from intention to implementation, through recruitment grounded in care, methods shaped by accessibility, analysis that honors complexity, and design that responds to what was truly heard.

It is important to remember that how we listen shapes what we build. When we move from extractive habits to inclusive practices, we uncover the friction that most needs our attention, the kind that can't be seen from the center. The work won't always be perfect or complete. What matters is that we keep showing up, again and again, with the willingness to adapt, even within constraints.

In the next chapter, we will move from research to interface, embedding accessibility into layout, contrast, navigation, and interaction, so inclusive intent shows up in every pixel, not just in principle.

Part 2

Designing for Accessibility

In this part, you will move from foundational theory into practical application, equipping you with the tools and techniques needed to build accessible digital products. These chapters explore how inclusive design principles translate into user interface design, content creation, web development, and mobile experiences.

You will learn how to apply accessibility best practices across visual design, interaction design, semantic HTML, CSS, ARIA, multimedia content, and mobile-specific platforms like iOS and Android.

Whether you're designing with color, structuring HTML, writing alternative text, or building for touch and voice input, this part shows how to create digital environments that work seamlessly for people with disabilities – and better for everyone.

This part includes the following chapters:

- *Chapter 5, Designing Accessible User Interfaces: Visual and Interaction Design*
- *Chapter 6, Creating Accessible Content: Writing, Images, and Multimedia*
- *Chapter 7, Developing Accessible Websites: HTML, CSS, and ARIA Best Practices*
- *Chapter 8, Mobile Accessibility: Designing for Touch and Voice Interfaces*

5

Designing Accessible User Interfaces: Visual and Interaction Design

By Nandita Gupta

Accessible user interfaces are key to creating experiences that work well for everyone, no matter their abilities or how they interact with technology. In this chapter, we will cover important aspects of designing with accessibility in mind, such as using color effectively, designing for cognition, supporting multiple input methods, and scaling accessibility across different devices and platforms. The goal is to help you build interfaces that are not only accessible but also intuitive and engaging for all users.

This information is valuable because accessibility is essential for inclusivity and usability. When you design thoughtfully, you remove barriers and create more meaningful, flexible experiences that reach a broader audience. This chapter gives you practical insights and strategies to make your designs truly inclusive and adaptable to real user needs, and covers how these strategies are beneficial for users with disabilities.

In this chapter, we will delve into the following topics:

- Using color within visual design
- Designing with cognitive disabilities in mind
- Designing for multiple alternative input methods
- Scaling designs for improved accessibility

Using color within visual design

Color is a powerful element of visual design. It can evoke emotions, guide user interactions, and create a memorable experience. However, with great power comes great responsibility. Designers must recognize that color perception can vary significantly across different people, cultures, and contexts. What might seem intuitive or universally understandable to one group of users can be confusing or exclusionary to another.

This section will explore how to use color effectively in a way that ensures accessibility for all users. We'll dive into color contrast, the use of patterns, shapes, and icons in place of color for information, and discuss how to design for cultural considerations in safety and warning systems. The goal is to create designs that cater to a broader, more diverse audience – one that includes the estimated 300 million people worldwide who live with color vision deficiencies (according to the World Health Organization).

Using color for time-sensitive decision-making and safety

Traffic lights offer a familiar example of how color is used to convey information quickly and instinctively. But for individuals with red-green color blindness – a common form of color vision deficiency – the distinction between those lights may not be clear, creating potential safety risks. This challenge becomes even more pronounced in high-stakes environments such as manufacturing, where split-second decisions can affect not only productivity but also worker safety. In such settings, relying solely on color to communicate urgency, status, or system conditions introduces unnecessary risk.

> The company RealPars wrote an article that showcases how color can make a huge impact within the manufacturing systems by using color indiscriminately throughout the system, where it not only becomes a cognitive overload but also a potential safety hazard: `https://www.realpars.com/blog/high-performance-hmi`.

Another important consideration is that the meaning of color is not fixed; it varies across cultures and contexts. For example, red is widely used in North America and Europe to signal danger or "stop," but in parts of East Asia, it can also signify celebration or prosperity. Similarly, white may represent cleanliness or safety in Western cultures, while in others, it is closely tied to mourning and death. These differences become critical in global products, where misinterpreting a signal could lead to confusion or delay.

Instead, combine color with additional indicators – such as shapes, patterns, text labels, or auditory signals – to ensure information is accessible to all users, including those with color vision deficiencies. For example, pedestrian traffic signals can be made more accessible by supplementing red and green colors with additional cues that don't rely on color perception. Instead of relying solely on color, do the following:

- Use text labels, such as "Stop" and "Go," displayed directly beneath or inside the lights. This helps users with color vision deficiencies and also supports understanding in unfamiliar environments. Audio cues, such as a spoken instruction or tone pattern, can further enhance accessibility.
- Pair colors with shapes – for instance, a red circular light for "stop" and a green square light for "go." This shape-based distinction reinforces meaning even when the color is not easily distinguishable.
- Incorporate symbols, such as a red figure standing still and a green figure walking. These provide an intuitive visual reference, especially helpful for users with low literacy or language barriers.

Another example of accessible color use at a global scale comes from international standards developed by the **International Organization for Standardization** (**ISO**). ISO provides a coherent set of graphical symbols designed to overcome language barriers and support consistent interpretation across regions. These symbols combine color with patterns, icons, and text to convey critical information clearly and efficiently. By using multiple redundant cues – rather than depending on color alone – ISO standards offer a practical model for designing inclusive signage and interface elements that work across diverse languages, cultures, and abilities.

> A solution that may be accessible for some might potentially be triggering for others. For instance, a flashing red light paired with a bold warning icon and an alarm sound can immediately grab attention and prompt quick action. On the other hand, the flashing lights might be triggering for users who experience photosensitive disorders.

Using color contrast to ensure legibility and clarity

Color contrast is essential for making sure your content is legible and accessible to users with low vision or color blindness. Contrast ensures that text stands out clearly against the background, making reading easier. Without sufficient contrast, users may find it difficult to differentiate between text and background, leading to frustration and increased cognitive load.

A dark gray or black text on a light-colored background (or vice versa) is ideal for readability, but it's important to verify that the contrast meets accessibility requirements. According to **Web Content Accessibility Guidelines** (**WCAG**) 2.1, standard body text must meet a minimum contrast ratio of 4.5:1 against its background to be considered accessible. For larger text – typically 18pt regular or 14pt bold – the minimum contrast requirement is slightly lower, at 3:1. These guidelines help ensure that users with low vision or color vision deficiencies can comfortably read content without strain.

Here is an example denoting the different levels of color contrast along with whether or not they would fail WCAG guidelines:

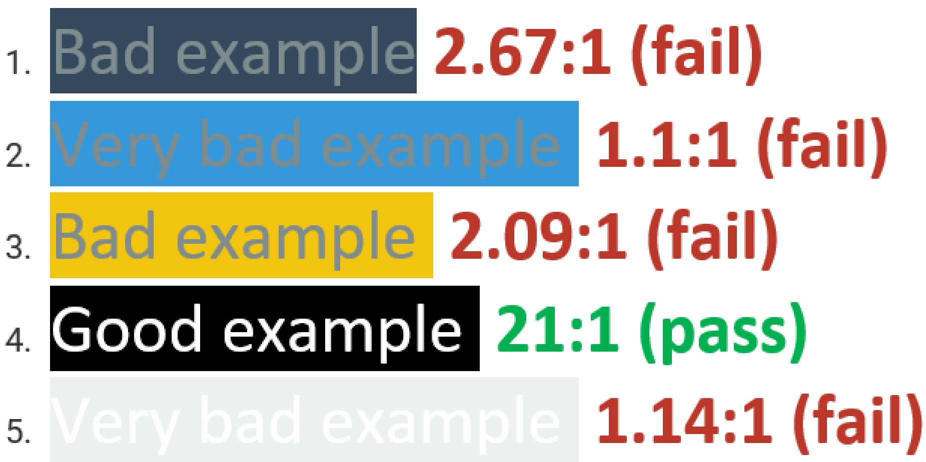


Figure 5.1: Color contrast vs. WCAG guidelines

To support readability, designers must evaluate color combinations early and often using tools that test contrast ratios against accessibility standards. Tools such as the WebAIM **Color Contrast Checker** make this easy by allowing designers to enter specific foreground and background color values and receive instant feedback on whether the combination complies with WCAG requirements. Integrating this kind of testing into the design workflow ensures that content remains legible not just in ideal conditions but across a wide range of visual abilities and device settings.

> **California State University, Northridge** (**CSUN**) provides a good article on good and bad color contrast, as well as a helpful contrast table: `https://www.csun.edu/universal-design-center/web-accessibility-criteria-color-contrast`.

Using color in charts and diagrams

Charts and graphs used in interfaces and web pages are another example where data-driven designs use color to differentiate between categories or highlight trends. However, relying on color alone to represent different data points is not sufficient, and using patterns or textures in addition to color can be valuable for perception.

For example, in a pie chart, each segment can be distinguished not just by color but by using solid fills, stripes, dots, or other patterns. These variations help users differentiate between data sets even if they can't perceive color accurately. When using legends, pair each color with a clear text label and a concise description of what that data point represents. This ensures the chart remains understandable regardless of visual ability. In addition, all charts and graphs should include descriptive text alternatives – either as captions, summaries, or accessible data tables – to support users who rely on screen readers or those with cognitive disabilities who benefit from written context.

Looking at the following pie chart, we can see that different colors have been used to differentiate the segments, along with texturing each slice, plus labeling each section with specific values:

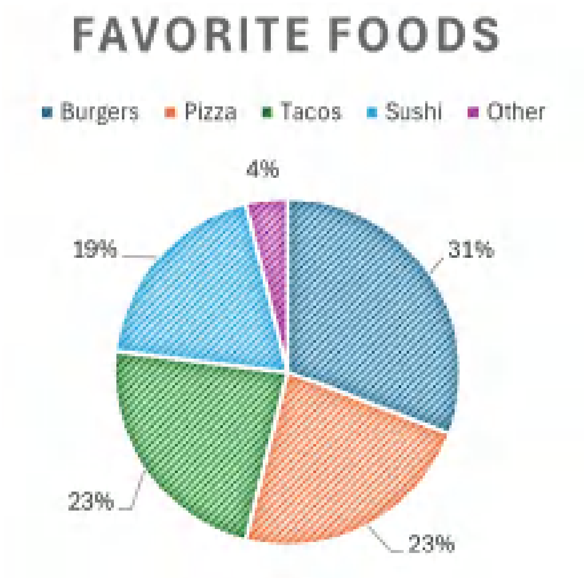


Figure 5.2: A pie chart with different colors and clear labels (though it could be improved further for better accessibility!)

To improve it, the colors used for the **Burgers** and **Sushi** labels are quite similar, so we could change one of the labels' colors. We could also use different textures for each slice; however, lots of different, clashing textures can be overwhelming for the consumer, so they should be used wisely.

Using color to convey errors

In web design, color is often used to indicate errors; for example, when filling out forms, green indicates success and red indicates failure. However, users with color deficiencies may not easily differentiate between the two, and one must not assume universal color perception for everyone.

About 8% of men and 0.5% of women worldwide experience some form of color blindness, which can make color-dependent information inaccessible. Color blindness varies from difficulty distinguishing between certain hues, such as red and green, to a complete inability to perceive any color at all (achromatopsia).

To ensure accessible form validation flows for visual design and color, do the following:

- Add icons alongside the colored text, noting success or failure. For example, a green checkmark and a red cross provide a clear visual cue about the status, regardless of the user's ability to perceive color.

- Use descriptive text labels to communicate validation states clearly. Text such as **Your email address is valid** or **Form submitted successfully** helps confirm successful actions, while error messages such as **Please enter a valid email address** or **Submission incomplete – please complete all required fields above** guide users toward resolution. These text labels also support screen reader users.
- Using a shape or layout, such as highlighting the border of a form field in error with a dashed or thick outline, helps users who can't perceive color differences.

Figure 5.3: Field validation utilizing symbols, text, and borders

- Provide audio alerts or haptic feedback for users relying on assistive technologies to signal errors without visual cues.

Using color to denote action and interaction

Affordance is a powerful design principle that signals to users what actions are possible, whether it's clicking a button, dragging a slider, or toggling a switch. These visual and interactive cues help users intuitively understand how to interact with an interface, often without needing instructions. When affordance is combined thoughtfully with color, it creates visual patterns that reinforce meaning and encourage positive, predictable interactions. Designing strong affordances reduces guesswork, guides user behavior, and supports users with cognitive or motor challenges by making the interface more intuitive and easier to navigate.

On both mobile and web platforms, buttons and links often use color to indicate action or interaction. However, if the button's text is not in sufficient contrast to the background, users with low vision may not be able to interact with the site effectively. Thus, it's important to ensure that buttons have high contrast with their backgrounds. A blue button with white text offers excellent contrast and ensures visibility across different devices and for users with varying vision abilities. Additionally, use hover effects to indicate interactivity, as well as use clear, descriptive labels that explain what action will be taken when clicking the button (for example, **Submit form**).

Thus, tangible ways designers can leverage affordance for accessibility include using clear labels on buttons, incorporating tactile or animated feedback for touch devices, ensuring interactive elements have distinct borders or shadows to stand out, and employing consistent iconography that users can learn and recognize. Combining these visual cues with screen reader-friendly attributes ensures that users relying on assistive technologies receive the same information through audio or haptic feedback, making the experience inclusive for all.

Color is more than just a design element; it communicates meaning. From traffic lights to websites, color is often used to convey urgency, status, or categorization. But color perception is not universal, and it's vital to understand the key differences in human perception of color. Using color for accessibility design includes various considerations such as contrast, use of text, and context. These elements, in addition to others, must be applied to create clarity, especially while designing for cognition.

Designing with cognitive disabilities in mind

Designing user interfaces for cognition and cognitive disabilities ensures equal access to information and functionality for all users. Cognitive disabilities, such as dyslexia, ADHD, and learning disabilities, vary widely, and complexity in design can overwhelm these users. The key to supporting all users is reducing complexity: making content clear, logical, predictable, and easy to navigate.

By embedding these into the design process, we create intuitive, predictable, and adaptable interfaces that empower users. These designs must also work at scale to build an inclusive digital ecosystem that supports diverse cognitive abilities.

Consistency across the user interface

Consistency and predictability are essential for supporting users with cognitive disabilities. When interfaces behave in expected ways, users can build mental models, recognize patterns, and anticipate outcomes, making it easier to learn and navigate digital environments. This aligns with one of Ben Shneiderman's *Eight Golden Rules of Interface Design*: "Strive for consistency."

This is closely aligned with WCAG 2.2 *Guideline 3.2: Predictable under the Understandable principle*. Within this guideline, *Success Criterion 3.2.3: Consistent Navigation (Level AA)* and *3.2.4: Consistent Identification (Level AA)* focus on ensuring that navigation mechanisms and elements with the same functionality are presented consistently across pages, helping users predict how to interact with the interface. This consistency reduces confusion and cognitive load, which is especially important for users with disabilities.

Consistency in design is crucial for general usability for all users, including users with disabilities, and can be implemented using the following guidelines:

- **Predictable patterns**: When patterns and behaviors are predictable, users can build familiarity with how things work, making it easier to navigate through complex systems or tasks.
- **Layout**: Maintaining a uniform layout by placing navigation elements, buttons, and interactive components consistently across pages helps users quickly learn where to find key functions. This principle is reinforced in WCAG 2.2, which introduces a requirement for consistent help mechanisms across pages (`https://www.w3.org/WAI/WCAG22/Understanding/consistent-help.html`).
- **Visual consistency**: Using the same icons, buttons, and colors throughout enables users to easily associate actions with their visual cues.
- **Interactions**: Predictable interactions are essential; sticking to common conventions (such as a trash can icon for deleting) prevents confusion and creates a smoother, more intuitive experience.

Taking a specific example, buttons that provide similar functionality should be consistent to alert the users to clickable elements and behaviors. The following figure demonstrates a good and bad example of button consistency:

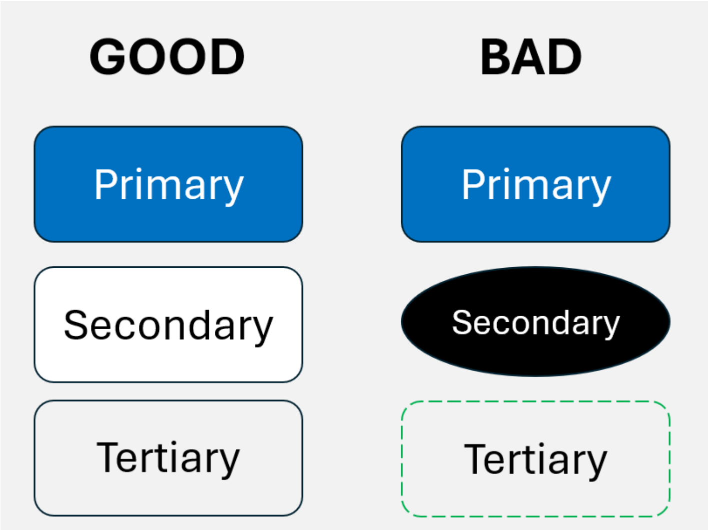


Figure 5.4: Good vs. bad button consistency

Once these foundational UX principles are put into practice, the next step is ensuring those design patterns work well across all users, including those with accessibility needs. The principles behind accessibility aren't one-size-fits-all; instead, design patterns should be flexible and adaptable. Here are three tips that can support accessibility:

- Many users with cognitive disabilities rely on keyboard shortcuts for faster navigation. Make sure that all interactive elements, such as buttons, links, and forms, are accessible through keyboard commands alone, and that the focus order is logical (take a look at the *Information architecture for accessibility* section at the end of the chapter for an example of good vs. bad focus order). It's also important to ensure those keyboard shortcuts and interactions are consistent throughout the experience.
- Provide contextual help or tooltips for complex interactions that require additional information. These should be accessible and easy to understand, guiding users without overwhelming them with information. But keep in mind that tooltips shouldn't be a substitute for bad design; the design should be easy to understand, while the tooltip provides extra information in a concise way.
- Form fields, buttons, and other interactive elements should always use clear, descriptive labels that communicate their purpose at a glance. Avoid vague or generic terms such as "Submit" and instead, use action-specific language such as **Submit Your Order** or **Save Your Work** to help users understand exactly what will happen next. Where appropriate, supplement these labels with brief, helpful guidance, such as input examples (**MM/DD/YYYY**) or field requirements (**Password must be at least 8 characters**) to reduce uncertainty and prevent errors.

Progressive disclosure and abstracting away unnecessary content

The key to building progressive disclosure and abstraction is not what you add to your user interface but what you hold off. Only the necessary information is presented at each step of an interaction, avoiding the need for users to memorize details about previous steps. Compare this to cooking a complex dish: a long list of ingredients followed by a chaotic and disorganized list of instructions. While it's important to provide all the necessary information, such as ingredients, steps, and materials required, presenting that information in a logical order can simplify the cooking process.

Similarly, for the user interface, keeping tasks simple and breaking down complex processes into smaller, manageable steps can ease the cognitive load. Here are some actionable ways to build progressive disclosure within a user interface:

- For tasks that require more thought or action (e.g., form submissions), break them down into smaller steps. This helps users stay focused and avoids overwhelming them with a long series of actions.
- Avoid overwhelming users with too many choices, too much text, or excessive visual clutter. Opt for clean, well-organized layouts with clear headings and labels.
- Give users an indication of how far they are in the interface/task. For example, if a user is filling out a long form, displaying a progress bar or step indicators can help them visualize where they are in the process, making the entire task feel less overwhelming. When creating the progress bar, use a combination of color and clear text (such as numeric percentages), and make sure assistive technologies such as screen readers announce the progress correctly.

It's important to note that in addition to supporting users with cognitive disabilities, the preceding techniques support a larger percentage of users, including people with intersectional identities and disabilities. Abstracting information and building clarity within a UX supports users with a wide variety of processing styles, from auditory to visual, and thus supports a larger percentage of users.

Easy reversal of actions

It's important that users feel in control and are not penalized for their mistakes. This is particularly important for users with cognitive disabilities, who may need more time to complete tasks and may make mistakes more frequently, and aligns with another of Shneiderman's *Eight Golden Rules*: "Support internal locus of control and allow easy reversal of actions." When users know they can undo or revise their actions without consequence, it reduces anxiety, encourages exploration, and makes digital environments feel safer and more supportive.

For example, offering a **Back** or **Cancel** option on every page allows users to correct mistakes without fear of causing permanent issues. This is especially important when it comes to data-sensitive and critical decisions, such as the deletion of important information, prompting users for confirmation, and ensuring the consequences of proceeding are clear.

Another example is when signing off on a critical task, such as authorizing a bank account change or completing a large money transfer online. In such cases, providing an easy way to review and reverse actions with a clear **Cancel** or **Edit** option not only reduces anxiety and errors for all users but is especially important for people with cognitive disabilities who may need extra time or multiple attempts to confirm their choices.

Alternatively, clear **Undo** or **Reset** options can also ensure users feel empowered to experiment without the risk of making irreversible errors. This is not just good practice for designing for cognition, but good UX practice. Designs that allow for recovery from mistakes are less likely to frustrate users and more likely to help them feel competent in their interactions.

Additionally, supporting error prevention and the undoing of actions (recovery) is essential by providing easy-to-understand error messages and simple recovery options. For example, instead of saying **Invalid entry**, say, **Please enter a valid phone number in the format (xxx) xxx-xxxx**. The expected format should be clearly stated within the instructions, and ideally, should communicate the expectations to the user upfront.

All these considerations support design for cognition that improves the overall usability of products. Applying these considerations across a multitude of input devices is crucial to ensure accessibility and user customization. In the next section, let's take a deeper look at designing for multiple input methods.

Designing for multiple alternative input methods

As technology evolves, so does the diversity of input methods users rely on to interact with digital interfaces. Users today use a wide plethora of technological input devices, from a keyboard or mouse to voice commands, gestures, or assistive technologies such as switches or eye-tracking devices to navigate the interfaces. To ensure your design is truly accessible, it's essential to create inclusive interaction patterns that support various input methods, offering a seamless and accessible experience for all users.

Interaction design needs to support multiple input methods, ensuring usability for people with motor disabilities, vision impairments, and other unique needs. In addition to a wide variety of technology, users have their own preferences, and thus, there is always a need for ongoing user feedback and iterative testing to refine these interactions over time.

The key to building an accessible user interface that works with multiple inputs is to test with a variety of assistive technologies, including eye-tracking and alternative input devices, to ensure your platform is fully inclusive. Users should be able to accomplish every task from browsing a page to completing a purchase without needing a traditional mouse or keyboard, and this is done by ensuring content and functionality are programmatically available to assistive technologies.

Voice commands to enable hands-free interaction

Voice commands have become an increasingly important interaction method for users with mobility impairments or those who prefer voice-based navigation. Designing for voice commands involves supporting both basic navigation (e.g., opening menus and selecting options) and more complex tasks (e.g., filling out forms and controlling multimedia content).

Within voice commands, multiple considerations come into play when users may be using assistive technologies such as Dragon or built-in OS features such as Dictation. It's important to build interfaces optimized for voice-based interaction and accessibility, integrating voice navigation across all critical user journeys and interactions to provide a fully inclusive experience.

To do this effectively, designers should keep the following considerations in mind:

- **Clear and consistent voice commands**:
    - Define a set of natural, simple commands that users can easily learn and use
    - Ensure voice commands are consistent and intuitive across the interface
    - Align command structures with common patterns used in similar products or platforms
    - Use familiar phrases for key actions (e.g., "Open menu," "Go home," or "Read aloud")
    - Allow for slight variations in phrasing (e.g., "Go home," "Go to Home," or just "Home") to accommodate different user preferences and speech styles
- **Feedback and confirmation**:
    - Combine voice-based feedback with visual or haptic cues wherever possible
    - Provide visual confirmation – such as highlighting a button or changing its state – after a command is executed
    - Reinforce interactions with multisensory feedback to help users verify that their actions were successful
    - Use confirmation cues to make voice interactions more accessible, predictable, and reassuring

A real-world example is using Amazon Alexa (`https://www.alexa.com/`), which supports voice-controlled navigation across a variety of devices, from smart home systems to streaming media. By offering clear, simple commands such as “Play music” or “Stop,” users can easily control their devices hands-free. Additionally, visual feedback via on-screen notifications or device feedback (e.g., **Music is playing**) ensures clarity. Personalized voice commands can be set up to control devices that eliminate the need for keyboard or button clicks.

Gesture controls to enhance navigation for touch-free interaction

Gesture controls have emerged as a powerful input method for users with limited mobility or dexterity. By using cameras, sensors, or touchscreens, gesture recognition allows users to interact with interfaces through movements such as swipes, pinches, or simple hand gestures. Designing for gesture controls requires ensuring that gestures are intuitive, consistent, and offer feedback to guide users through their tasks:

- It’s important to keep gestures simple, familiar, and easy to learn, so as not to cause cognitive overload. Complex or non-standard gestures can be frustrating for users with motor disabilities; thus, stick to common gestures such as swiping left/right, pinching to zoom, or holding a gesture for activation.
- Interfaces and systems must also provide visual or haptic feedback when a gesture is recognized. For example, a swipe gesture might change the screen content or trigger a page transition, and a clear animation should confirm that the gesture was successful.
- In many cases, it’s valuable to add personalization, as some users may find certain gestures physically difficult or uncomfortable. Allowing users to customize gesture mappings or substitute gestures with alternative inputs, such as on-screen buttons or voice commands, ensures that the interface adapts to their needs rather than forcing a one-size-fits-all solution.

A real-world example is integrating Cephable (`https://cephable.com/`), a service that allows users to enable voice- and gesture-based interaction using their existing inputs. Users can control their device with simple gestures, such as raising a hand to pause a meeting or swiping to change channels. By combining intuitive gestures with clear visual feedback (such as screen animations or color changes), this provides an accessible experience for people with limited dexterity.

Assistive technologies, including support switches, eye tracking, and more

Assistive technologies such as adaptive switches, eye-tracking devices, and sip-and-puff systems are vital for users with severe motor disabilities. These technologies offer a means of interacting with digital systems when traditional input methods are not viable.

It's crucial to ensure your platform is compatible with these devices, allowing users to navigate your site or app effectively. In addition to the previous information regarding voice controls and gestures, it's important to allow users to map actions to specific switch inputs or eye movements. For example, a user with limited hand mobility might use a switch to perform a mouse click or a keyboard key press. All interactive elements must be reachable and can be activated using assistive devices, including a logical tab order. The design must ensure the focus indicator is visible and easily navigable, and allow actions to be triggered via single or multiple inputs.

Apple's Switch Control is a system that allows users to control their iPhone, iPad, or Mac with a variety of external switches (`https://support.apple.com/en-us/118667`). Users can select and interact with on-screen items by pressing switches that are mapped to specific actions. This allows those with severe mobility impairments to use Apple products effectively.

Designing for multimodal inputs – such as voice, gesture, keyboard, and assistive devices – is essential for creating accessible experiences. However, it's not enough to apply these techniques in isolated instances. To truly support a diverse range of users, it is critical to be able to deploy and maintain all these techniques at scale, which is what we will look at next.

Scaling designs for improved accessibility

Designing for accessibility at scale requires creating a flexible and inclusive design system that works across a range of user needs and devices. This approach ensures that your platform remains usable and accessible as it grows. A well-designed system goes beyond visual elements, considering usability, cognitive clarity, and consistency across all interactions. It includes elements such as reusable components, font readability, user customization, and keyboard navigation, ensuring that the platform is usable by everyone, regardless of ability.

Building a modular, accessible design system

A scalable design system is crucial for accessibility. It involves creating reusable components (e.g., buttons, forms, and modals) that are thoroughly tested for accessibility. These components should be flexible enough to assemble into more complex designs without sacrificing accessibility, and this forms a sustainable foundation for an accessible platform.

One way to implement scale is by using reusable components that adhere to accessibility standards, such as minimum contrast ratios and compatibility with screen readers, and guidelines on how to implement these components. The design system must also provide clear documentation on how to implement these components, including how to add alt text, structure content for screen readers, and adhere to touch target sizes for motor disabilities.

These pieces ensure that, as the platform evolves, new features will be integrated with the same focus on simplicity, consistency, and cognitive support. This applies to components such as buttons that meet accessibility standards (e.g., proper contrast and screen reader labeling) and are reused consistently across the platform, as well as interaction and navigation patterns that are set up for the whole platform.

Elements that should be accessible at scale

Key UX elements such as readability, clear language, and user-friendly navigation form the backbone of accessible design. The way these elements support scale is by ensuring that the changes are only made once across the design system and UI toolkit. Thus, every designer or developer using these elements automatically bakes accessibility into their design.

Here are some key design considerations for elements:

- **Font readability**: Prioritize large, clear fonts with sufficient contrast to enhance legibility across devices and user conditions. Avoid decorative or overly stylized fonts that can hinder readability. Instead, choose simple, clean sans-serif typefaces – such as Arial or Helvetica – that scale well and maintain clarity on a variety of screen sizes and resolutions.
- **Color use**: To ensure clarity and inclusivity at scale, always supplement color with additional visual cues, such as icons, patterns, underlines, or text labels.
- **Guided focus**: Use consistent visual cues – for example, a clearly visible focus ring around active elements such as buttons or input fields – to help users maintain orientation while navigating. Following the WCAG AAA recommendation, this focus indicator should be high-contrast and persistent, ensuring it remains visible across all pages and components.

- **Clear language**: Use simple, direct wording and avoid jargon that might confuse users, especially those with cognitive challenges. Establish clear language guidelines addressing tone and readability levels to ensure content remains consistent and easy to understand across the platform.
- **Time-based interactions**: Provide users – especially those with cognitive disabilities – with the option to adjust or disable time limits on tasks or interactions, enabling them to process information at their own pace. To support accessibility at scale, implement these controls consistently across all relevant areas of your platform, ensuring users encounter a predictable and accommodating experience regardless of context.

Empowering personalization through user customization

Allowing users to customize their interface can significantly improve accessibility. A modular design system creates a foundation for personalized experiences, where users can adjust the platform to meet their unique needs, such as changing font sizes, colors, or layouts. The key to personalization is building responsive and dynamic user interfaces that let users control font sizes and color contrast to suit their preferences or needs.

A flexible design must work across different devices, from desktops to mobile phones. It should support zooming and text resizing without losing functionality or breaking the interface. Consistency across operating systems plays an important role in building an inclusive experience. This includes a clear, consistent navigation that works across devices and zoom levels. This concept ties to WCAG 2.2 *Success Criterion 1.4.10: Reflow (Level AA)*, which refers to the ability of content on a web page or app to adapt dynamically, adjusting layout, text size, and positioning so that users can view and interact with it without needing to scroll horizontally or zoom excessively, even when the display size changes (such as on mobile devices or when zoomed in).

Layout is a vital consideration. The design system should allow customers to rearrange sections and ensure that content adapts seamlessly to various screen sizes and orientations. A real-world example is Webex, which allows users to customize their layout within the meeting experience to be able to pin their sign language interpreter, change stage view, hide attendees without video, and many more. This empowers users to be able to customize their meeting experience for an inclusive experience (`https://blog.webex.com/collaboration/fostering-inclusive-and-accessible-communication-with-webex-suite/`).

Personalization also extends to language and localization. When scaling globally, it's essential to support language customization and localization, which can impact layout, content structure, and focus behavior. Some languages (e.g., Arabic) read right-to-left, so the layout must adapt accordingly. Others, such as German, may require more space due to longer word lengths. Additionally, it's important to maintain clarity and consistency when localizing content, especially for users with cognitive disabilities.

Information architecture for accessibility

While accounting for information architecture, account for multi-modal inputs, including keyboard-friendly navigation systems. For users with motor disabilities or those who prefer keyboard navigation, using the *Tab* and *arrow* keys, make sure that all interactive elements are accessible via the keyboard. This includes fixing tab orders and visual focus indicators to provide users using the keyboard and other input devices with a seamless experience.

It is valuable to organize the tabbing order so that it flows logically, from top to bottom or left to right. All the active elements on the page should be clearly highlighted (e.g., buttons and form fields) so users can easily track their focus, and include keyboard shortcuts where appropriate, ensuring they are well documented and customizable.

Looking at the following figure, although a visual reader might be able to discern the hierarchy of buttons and information, naturally moving their attention to the **Apply Now!** section, the tab focus order would create a chaotic experience for those just using a keyboard or relying on a screen reader:

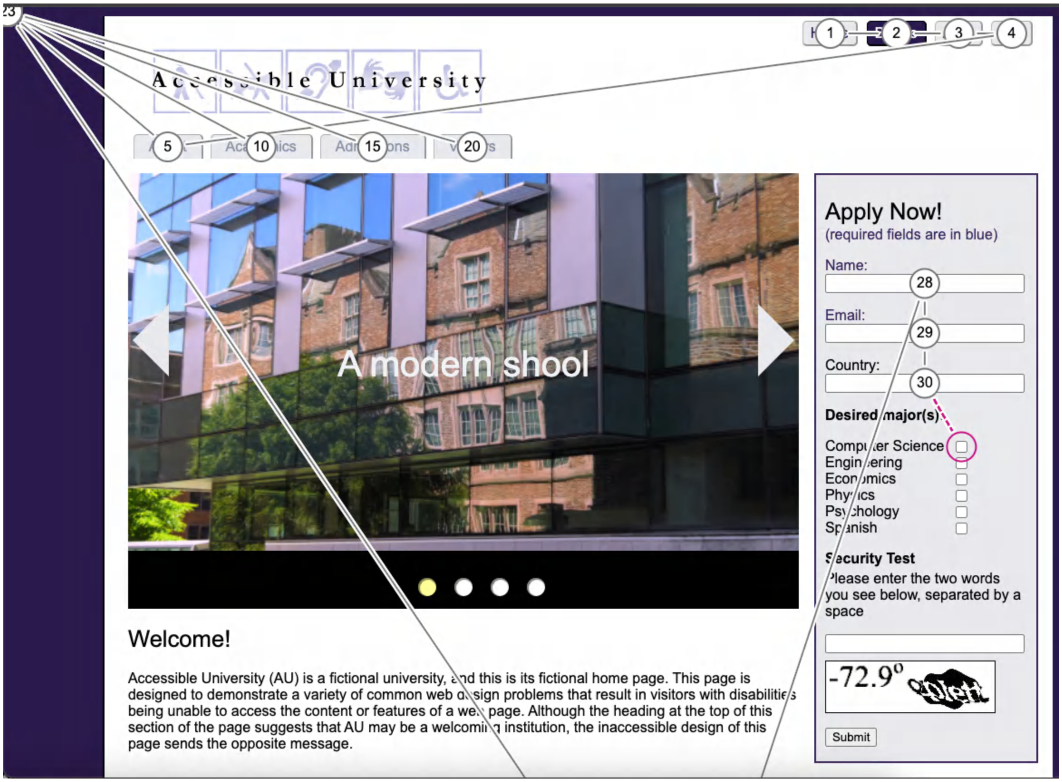


Figure 5.5: An example of bad tab focus order

However, in this next figure, the tab focus order has been fixed to create a more natural, easy-to-follow pattern:

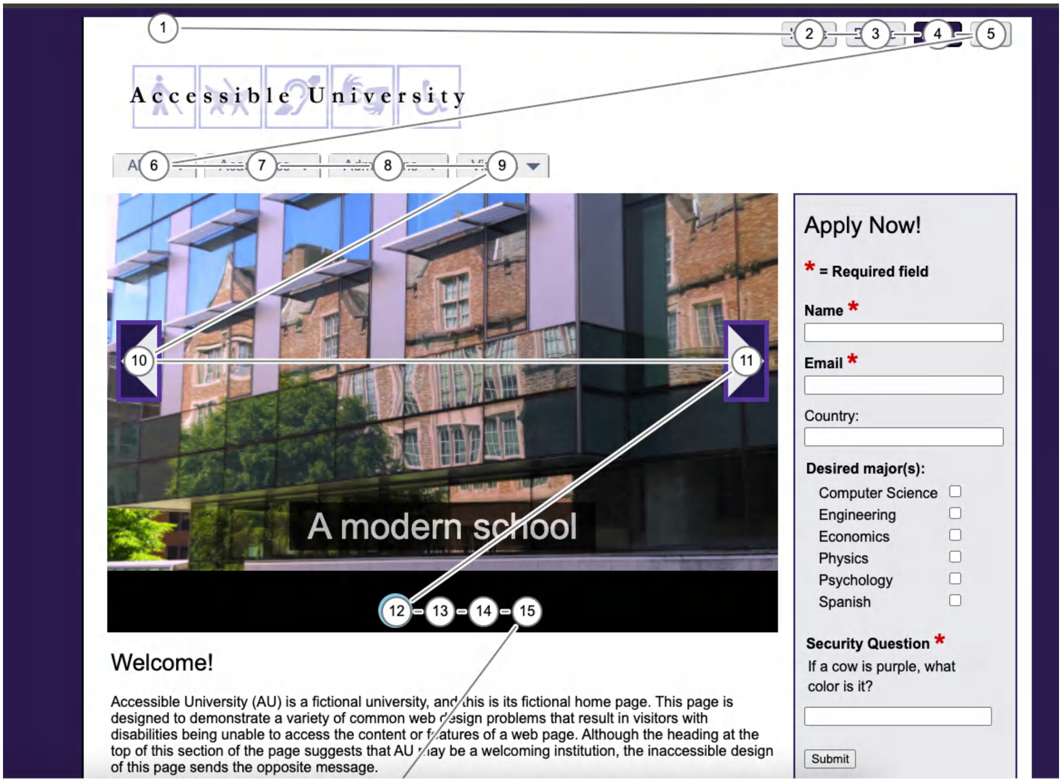


Figure 5.6: An example of good tab focus order

Summary

In this chapter, we learned how to design accessible user interfaces by focusing on key visual and interaction principles. Topics covered include using color thoughtfully within visual design to ensure clarity and inclusivity, designing for cognitive differences to reduce barriers, supporting multiple alternative input methods for diverse users, and approaches to scaling accessibility across various devices and platforms. These insights provide a solid foundation for creating interfaces that are both usable and accessible to a broad range of people.

This information is valuable because accessibility is a critical part of responsible and inclusive design. By understanding how to apply these principles, you can build products that empower users of all abilities, reduce frustration, and improve overall user experience. Designing with accessibility in mind not only meets legal and ethical standards but also expands the reach and impact of digital solutions and impacts revenue and adoption.

In the next chapter, you will learn about creating accessible content, specifically focusing on written content, images, and multimedia.

6

Creating Accessible Content: Writing, Images, and Multimedia

By Jennifer Chadwick

It's been said that content is king, and that is true. Content in all its forms is used to tell a story, provide information, or sell a product. The message must be communicated clearly to be meaningful, understood, and acted upon. As a writer, you are creating the message.

In this chapter, we'll cover how to create accessible content that reaches as wide an audience as possible – one that includes people with disabilities. We'll look at the content types of text, images, and multimedia, and which formats of these are best for people's needs. We'll discuss the importance of using plain language when writing information, labels, and instructions. I'll provide you with guiding principles to keep your writing style inclusive for all.

At the end of the chapter, I'll share a Needs Walkthrough exercise you can do as a writer to test-drive your content against key use cases for people with disabilities. For that, consider gathering your **user experience** (**UX**) and visual **user interface** (**UI**) designers to walk through a prototype together.

This chapter covers the following main topics:

- "The medium is the message"
- Reviewing key actions to build an inclusive approach to content
- Making different types of content more accessible
- Conducting the Needs Walkthrough exercise

"The medium is the message"

That's a quote from Canadian philosopher Marshall McLuhan. In 1964, he coined this phrase in the first chapter of his book *Understanding Media: The Extensions of Man*. He meant that the way information is delivered (the medium) has a greater impact on society than the actual content (the message) itself.

Well, people with disabilities are part of society. If the way the message is delivered excludes some people from receiving the message itself, that "greater impact to society" will be negative. Put simply, if the medium is obstructing the message, you've got a problem.

McLuhan's observation has remained relevant to the evolving digital age for more than 60 years. Arguably, it's a philosophy that captures the critical importance of inclusive content design. We want to include people with disabilities in the audience the message is reaching. Their disability should not be a barrier to reception. Rather, the medium or, as we call it, the content format, can be changed to remove those barriers.

The phrase "The medium is the message" makes us stop and think in three stages:

1. **What information – the message – is being conveyed in the content?** Think carefully about the key messages you're trying to get across in your writing, image, or multimedia content.
2. **What content format have I chosen to share this information?** Look at your content choices. Are you using text, or a mixture of images and text? Have you been asked to use a video or audio clip?
3. **Does the content format I chose reach the widest audience and/or exclude anyone?** If you're using a video to explain something, is the same information available in a simple text format as well? In the same way, could that complex data table be written as a plain-language paragraph?

To create inclusive content, ask yourself these questions. These questions can also be leveraged when conducting the Needs Walkthrough exercise later in the chapter.

Next, I'll establish some key actions you can take in your daily work. These will help you strengthen your skills by maintaining an inclusive mindset in your approach to content writing and format choices.

Reviewing key actions for building an inclusive approach

Let's look at some fulfilling and actionable ways to build out an inclusive writing practice. Here we will look at some more general ideas, and then later, we will look at content-specific methods.

Familiarize yourself with user needs

To create content that is free of barriers for people with disabilities, you need to familiarize yourself with the way they will experience it. The W3C **Web Accessibility Initiative** (**WAI**) website has a page called Diverse Abilities and Barriers (`www.w3.org/WAI/people-use-web/abilities-barriers`).

It's an excellent resource for understanding the personal needs and preferences associated with different disabilities. It provides engaging user stories of people as they work, learn, and shop.

It outlines the digital content and design barriers for five disability types:

- Auditory (e.g., deafness, deaf-blindness, hard of hearing, etc.)
- Cognitive and learning (e.g., autism, ADHD, memory loss, learning disabilities, etc.)
- Physical (e.g., mobility issues, quadriplegia, amputation, seizures, etc.)
- Speech (e.g., stuttering, muteness, and apraxia (inconsistent articulation and sounds), etc.)
- Visual (e.g., blindness, limited or low vision, deaf-blindness, color vision deficiencies, etc.)

Keep multiple user experiences in mind

Content and user interface designers must remember that they are not the end user. Your personal needs, perceptions, and abilities may differ from the wide diversity of users of your product or service.

Designing primarily with yourself in mind, especially if you don't have a disability, will inevitably exclude several people from experiencing it. Instead, start with the resources published by the WAI within the **World Wide Web Consortium** (**W3C**). They're practical, effective, and map directly to the **Web Content Accessibility Guidelines** (**WCAG**), a standard used as a basis for global accessibility laws, acts, and regulations.

Learn from real users whenever you can

The best way to learn is from real people and their lived experience. Invite people with disabilities to collaborate and co-design with you on a project. Keep collecting end-user experiences through interviews, surveys, prototype walkthroughs, and group discussions. This practice is exciting, effective, and beneficial for everyone. You will validate your new knowledge by observing and asking people how they do things and how they use assistive technology and adaptive strategies to get things done. Learning from people directly also discourages you from making assumptions about "all blind users" or "all users with ADHD."

If budget or time constraints won't allow consultation of people with lived experiences, use the list of W3C's five disability types to create a checklist, and 1-2 user scenarios or tasks for each type.

Then, walk through the experience of interacting with your content sample or design prototype. Was the user able to complete the task or experience the content or design without confusion or difficulty? If not, make notes to improve the experience or remove barriers.

Write all content in plain language

The International Plain Language Federation says that "communication is in plain language if its wording, structure, and design are so clear that the intended readers can easily find what they need, understand what they find, and use that information" (`https://www.iplfederation.org/plain-language/`). It doesn't matter whether your subject matter is complex (medical terminology, technical instructions, or legal terms) or simple (information in a blog post). Writing in plain language supports reading comprehension at all levels to reach the widest audience.

Nielsen Norman Group, a leading user experience research and design expertise firm, conducted a usability study in 2017 with domain experts in science, technology, and medical fields. The question posed: does plain language dumb down highly technical or academic content? The study discovered that "even highly educated online readers craved succinct information that is easy to scan" (`https://www.nngroup.com/articles/plain-language-experts/`).

The UK Government also conducted research that found 80% of people prefer reading content in plain language. Mark Morris was Head of Clear English at the UK Government's Department of Health and a former speechwriter for the Health Secretary. In a 2014 blog, he coined a phrase that speaks to an evolution in the way content authors have embraced plain-language writing. "Clarity is king."

In 1994, Microsoft co-founder Bill Gates saw the power of online content creation on the World Wide Web to reach a wide audience. He said, "Content is king." Now, it's clarity that matters most.

But you don't have to be the author of a complex medical journal, a speechwriter for the Ministry of Health, or even Bill Gates to know that plain language is useful for all users.

> Later in the chapter, in a section called *Text*, we'll cover specific plain language tips for writing paragraphs, labels, and instructions.

Familiarize yourself with barriers and alternatives

You now have knowledge of disability needs and are ready to apply the plain language principle. But let's face it, certain digital content media (formats) are not inherently accessible to people with varied disabilities and needs. For example, you may choose to present a news story in a video, but some people with visual impairments cannot perceive the screen easily, or at all.

Does this mean you can no longer use video to convey information because some people can't experience it? Of course not. Use the video format, but ensure it's paired with a media alternative, such as a transcript. Transcripts present the title, character descriptions, spoken dialog, sounds and actions on-screen. It's the same information, just accessed in a different way.

In this section, we've learned that if you choose to use a content format that you know inherently won't work for some users, you must combine it with an alternative format that will reach the rest of your audience. Next, we'll look in depth at each content type and which alternatives work best for users with disabilities.

Making different types of content more accessible

The WCAG standard is structured around four principles to support meeting the needs of end users. The first three are:

1. **Perception**: "I can access the content, recognize what it is, what it means and what to do with it – without any barriers."
2. **Understanding**: "I understand the purpose of the content, its message or instructions clearly enough to take action."
3. **Operability**: "I can complete the action, based on the label or instruction provided in the content."

> For those wondering, the fourth principle is **Robust**: "This website or application can be interpreted by a wide variety of user agents, including assistive technologies that people use, like screen readers, keyboards, switch controls, voice commands, etc." However, this is mostly related to coding and implementation, which we are not considering here.

In the following sections, we will look at different types of content and see how they can be made more accessible.

Text

Text is a universal content medium and generally the most accessible. Users with visual impairments, learning difficulties, memory loss, and/or attention deficit disabilities use screen reader software to announce content on a page. Text can be parsed by a screen reader (such as JAWS, NVDA, VoiceOver, and Narrator) and other reading tools built into software without issue. When PDFs are saved properly, the text within can also be parsed by the same tools.

Text informs, instructs, and identifies. Text falls under the Perception principle because a user can access and perceive the text either visually or using screen reader software.

In the following sections, we will look at how to make text more accessible through "scannability," as well as more principles to provide plain language.

Make pages easy to scan

In 2008, Nielsen Norman Group (NN/g) conducted an eye-tracking user research study on the web (`https://www.nngroup.com/articles/how-little-do-users-read/`). They found that, on the average web page, users have time to read, at most, 28% of the words during an average visit; 20% is more likely.

For another NN/g study on mobile in 2010 (`https://nngroup.com/articles/mobile-content-is-twice-as-difficult-2011`), researchers at the University of Alberta found that reading comprehension was impaired when content was presented on a mobile-size screen versus a larger computer screen. A simple explanation for this result was that, with a small screen, users saw less of the text at any given time, so they had to rely more on their memory to access contextual information needed during reading. In other words, the smaller screen resulted in a higher working-memory load. People could not sustain that higher load, so their comprehension suffered.

For this reason, you can optimize the experience for everyone by providing precise and concise content. To help make your content easier to scan, particularly for people with cognitive needs, learning disabilities, or low vision, consider the following advice on text formatting.

Headings

The purpose of headings is to introduce the relevant content underneath them.

For your web page, create a nested structure with headings that introduce each area of content. Define the main heading subject and then use sub-headings related to your point that are "nested" below. The front-end developer will use HTML heading tags to create your nested structure.

For example, your key heading will be heading Level 1. From there, the document or page can have multiple subheadings at Level 2, and underneath each of those, Level 3 subheadings, and so forth.

Having content broken up into clearly defined sections makes it easier to scan.

Screen reader users can press shortcut keys to jump to content (headings, links, graphics, tables, etc.). In this case, pressing the *H* key announces each heading.

> Remember not to overuse headings. Users with visual impairments or learning difficulties may use text-to-speech software such as a screen reader to parse the content onscreen and gain an understanding of the page's structure. Overusing headings can lead to confusion, so make sure to only include them when relevant.

Paragraphs

When writing paragraph content, it's important to use clear, concise language. Also, consider keeping the paragraph length to four or five sentences.

Breaking up text into short paragraphs is called chunking. This makes the text easier for the user to scan as their eyes move down the page, or with a screen reader, as the speech output will pause after each paragraph. It avoids a dreaded "wall of text," which can be overwhelming. It reduces eye strain as well as cognitive overload. It improves language processing for people with cognitive disabilities, those with memory impairments, or anyone multitasking.

> Avoid using ALL CAPS. To a visual user, using all capital letters for your text looks like the message is being shouted. Screen readers don't pick up on the text emphasis, so the impact is lost on non-visual users anyway. Also, all-caps text reduces the distinctive shapes of words, making it harder for people with dyslexia to recognize and process them quickly.

Bulleted and numbered lists

Bulleted and numbered lists improve accessibility by making content easier to read and navigate, especially for people using screen readers (as it will announce that the content is a list), those with cognitive disabilities, and individuals who benefit from structured information.

Using lists helps break up complex ideas into digestible points – however, use a bullet list when the items don't need to be in a particular order, and use numbered lists when providing a sequence, such as instructions.

Navigation menu and button labels

Navigation menus are generally a structured set of links. Therefore, your link text should be a clear description of the destination page, e.g., "Who We Are", "Our Services", or "How [the product] works". Sometimes, though, a navigation menu will use buttons as well.

Keep these menu links and button labels clear and concise. This is essential for all users to understand the structure of a website, mobile app, or document.

Vague or overly complex labels can be confusing to all users, but especially those with cognitive disabilities and non-visual users who rely on screen reader output. Labels on buttons and link text should tell the user instantly what the controls do, how they function, and the expected outcome.

To help, use descriptive labels in plain language and avoid jargon. For example, though "Get in Touch" is common, use "Contact Us" instead. This will optimize understanding and leave no doubt as to where that link or button is taking them.

Navigation landmarks and menu links

Ideally, pages are constructed with landmarks (or regions) to define their structure. Landmarks include header, footer, main body, and navigation, where your navigation menu sits.

Developers implement code so that screen readers announce the landmark type. However, if there are two navigation menus on a page, your role is to write a short description to distinguish them. For example, "Global navigation menu" and "North America services menu".

Links

You might use a link as part of a sentence in a paragraph. This gives users a quick way to visit another page and learn more information. In this case, don't use phrases such as "Click here" or "Learn more" for the link text in your sentence.

For starters, not all users click with a mouse. But also screen reader users may use shortcut features that extract a list of all the links on a web page. This takes them out of the context of the sentence, so if you're using generic terms such as "Learn more" all over the page, the user will simply hear a long list of "Learn more" links without any differentiation between them. Instead, create a hyperlink by using the words that describe the subject of your sentence or paragraph. For example, with the sentence "Learn more about our accessibility features," hyperlink the words "accessibility features," or make the entire sentence a link.

Labels for buttons and interactive controls

Vague or overly complex labels on buttons (or any interactive controls) can be confusing to all users, but especially those with cognitive disabilities and non-visual users who rely on screen reader output. Labels should tell the user instantly what the controls do, how they function, and the expected outcome.

As per the guidance for links, avoid using "Click here" or "Learn more" for button labels if you can. If you do so, you must provide the developer with an extended but concise description for screen readers – for example, "Learn more about the Tier One Dental Plan" using `aria-label`. Or, `aria-labelledby`, which uses existing content on the screen, such as a heading.

Form field labels and instructions

When creating forms, you want to make sure the form itself has a description and the form field labels and instructions are as accessible as possible. Here are some best practices for creating labels:

- The form itself should have a name or description, i.e. "2025 Cast Audition Form." Developers will include this for screen reader users using either `aria-label` or `aria-labelledby`.
- Form field Labels should be concise, specific, and placed near the input field.
- Use sections and headings to organize information logically. As a writer, you can work with the UX/UI designer on the structure.
- Indicate required fields clearly using an asterisk (*) and ensure the developer codes the form field using `aria-required`, so the word "required" is announced to screen reader users. Conversely, if most form fields are required, too many asterisks can be

overwhelming. In this case, write an instruction: "All fields are required unless marked optional."

- Show clear, specific error messages next to the relevant field. Describe exactly what needs to be corrected and actioned, in that moment, before the user can proceed with the form.
- Ideally, put the expected input formats in your label (e.g., MM/DD/YYYY for dates). Screen readers will announce the label and format instruction at the same time. This helps low-vision users zoom in with screen magnification – they can see it all at once.
- Import formats (MM/DD/YYYY) can also be placed as helper text below the field and announced to screen reader users with `aria-describedby`.
- Don't put important instructions in placeholder text. Make sure instruction and formatting hints (e.g., MM/DD/YYYY) are part of the label.

And here are some best practices for instructional text:

- Use simple and familiar language, avoiding jargon and complex wording
- Instructions should be written in an active voice, clearly directing what actions to take, in the order they need to be done
- Ensure essential user information is placed before the start of the form, e.g. "You must be 18 to apply for this credit card"

Group related form inputs under headings

Group thematically related fields visually and programmatically, such as name fields distinct from address fields. This helps to break long forms into more digestible sections.

Data tables

When using data tables, consider these best practices:

- Ensure tables are used for tabular data and not just for structuring page layout.
- Provide column and row headers for clarity.
- Avoid merging or splitting cells, which can confuse screen readers.
- Offer CSV or Excel downloads for alternative access.
- Provide alt text to describe any instructional images or SVG files in the table to screen readers. If the images or SVGs are decorative only and aren't necessary as part of the data, they should have no alternative text. The alt attribute remains empty, with no space in between, i.e., `alt=""`.
- Use labels for interactive controls such as buttons, toggles, menus, lists, and information icons, as you would for these items outside the table.

There are also code and visual design practices that you won't be responsible for, but can support your content in a data table. When working with developers and designers, remind them to do the following:

- Code the table to ensure that tables are keyboard navigable and compatible with screen readers
- Create a visual style with alternating row colors for better readability

The best practices support blindness, as screen reader software parses tables by announcing the column header each time the user moves to a cell in each row in a new column. This ensures users aren't lost, and the data relationships are intact.

Some users with low vision use magnification software to zoom into a screen, up to 800%. Having a concise table with all the information contained is good. They don't have to move around the page to find information. Users with learning difficulties can pick up on relationships in tables more easily sometimes than reading a text paragraph.

> As mentioned, some of the best practices relate to the design and coding of the table itself, which is the responsibility of the visual UI designer and developer. However, as the content author, you may well be choosing the format (in this case, a table). So, it's important to be familiar with best practices and how visual presentation, style, and code implementation impact certain users with disabilities.

Plain language principles for text

So far, we have talked about the formatting of text on the page or screen to make it easier to scan. Now let's discuss how you write the text itself and communicate the message.

Front-load your sentence with the subject

Since the early days of journalism, the first sentence of a news story should answer the central 5 Ws – who, what, when, where, and why. Introduce your subject and topic first, and make your point about them at the end of the sentence. This way, the context is clear from the beginning.

Be concise

In his 2001 book, *Don't Make Me Think*, Steve Krug emphasizes the importance of using clear, simple language in your writing and for instructions and labels. This way, users don't have to think – spending mental energy deciphering what your labels or content means.

His other key guidance is to "Get rid of half the words on each page, then get rid of half of what's left." With a combination of on-point sentence-writing, plain language, and brevity, you can achieve all you want to say in the fewest words possible.

Write in an active voice

Use active verbs rather than passive verbs when writing:

- Passive verbs are when the sentence's subject has something done to it. For example: "Several children's charities have been supported by Company B since it went into business 10 years ago."
- Active verbs are when the sentence's subject does something. For example: "Company B supports several charities. It began its support over 10 years ago and continues to help children in need."

As you can see, the active voice is clearer and more engaging to the reader and speaks more confidently about the subject.

Use simple words

From a 2023 survey, 51% of Canadians are at reading level 3 or above – general literacy. However, 19% of Canadians scored at level 1 or below (`https://decoda.ca/insights-from-canadas-adult-literacy-survey/#:~:text=Key%20Findings%20from%20the%202023%20Canadian%20Survey&text=Literacy%3A%2051%25%20of%20Canadians%20are,at%20Level%201%20or%20below`).

In the US, approximately 54% of adults read at or below a sixth-grade level (`https://en.wikipedia.org/wiki/Literacy_in_the_United_States`).

In the UK, the average reading level is generally considered to be around year 5, which is equivalent to a child aged 9 in the UK education system (`https://pmc.ncbi.nlm.nih.gov/articles/PMC11480634/#:~:text=Readability%20matters%20because%20it%20is,%E2%80%90old%20children%20%5B14%5D`).

Based on these statistics, don't use a complex word when a simpler one will do. Using the simplest terms in your language will allow your content to reach the widest audience.

Use a readability tool

As you write, you can use an online tool such as Hemingway App (`https://hemingwayapp.com`). It runs a scan of your text for its readability based on academic grade level. Aim for grade 6 reading comprehension, or lower (based on North American school systems). Alternatively, Grammarly (`https://www.grammarly.com/`) is a well-known AI-based tool that plugs into other software programs such as MS Word to scan and offer grammar corrections and suggestions for plain language.

Format combinations for text

Remember to combine content formats to keep content accessible to users with different needs. For example, if you use a data table, include the same information in a plain-language paragraph where you explain the relationships between the data – two ways to digest the same content.

> As you move through the rest of the chapter, you will see how text can be combined with images, video, and audio.

Let's wrap up our discussion about text by reminding ourselves what Marshall McLuhan said: "The medium is the message." Keep your messages conveyed in text plain and simple. This way, it can be perceived, understood, and operated upon.

Images

Let's look at different types of images that a designer may use and that you might collaborate with them on. Some image types contain words as well as graphics to convey meaning.

But first, it's important to learn why, when, and how to use a text alternative for visual content such as images and icons.

Types of images

When it comes to images, we can categorize them into four types:

- Decorative images serve a purely aesthetic function and do not add meaningful content, for example, a flowery background pattern on a website. These images do not need alternative text. The developer will keep the `alt` attribute, but it must be empty with no space, like this (`alt=""`). This means screen readers ignore them, preventing unnecessary distractions for users with visual impairments.

- Informative images are ones that convey meaningful content or information within them. A common example is step-by-step photos or illustrations on a recipe website, showing how to make a cake from scratch. The purpose of these images is to convey information in visual form. They need alternative text to describe them for non-sighted users. There may be text on the page or screen that relates to the image, but alt text should still be used to describe exactly what's in the image. Also, the alt text should not simply repeat contextual text around it that says the same thing.
- Functional images are images used to initiate actions, rather than to convey information. They are used in buttons, links, icons, and other interactive elements. For this reason, the alt text should describe the function, not the image itself – for example, a print button uses only a print icon to visually convey its purpose. Its alt text would be "Print" not "Printer." Similarly, the search icon is a functional image for a button, labelled "Search" not "Magnifying glass."
- Complex images are informative images that provide a lot of detailed and sometimes interconnected information or data points. These certainly are not just decorative. The data is usually displayed in a highly visual format such as pie charts, bar charts, diagrams, or infographics. There can be text mixed with color-coded lines, symbols, and shapes. In this case, the information is presented in a non-linear fashion, which makes it difficult for screen readers to navigate and share the information in a meaningful sequence to users with visual impairments.

Let's take a closer look at alt text, and why it is important. We will also look at why we should avoid using text in images.

Utilizing alt text

Working with a designer, you may choose images, photographs, or illustrations to accompany your writing piece. But if you do use them, it's your job as a content creator to transform their subject, purpose, and emotions in a way that supports people with blindness, limited vision, and/or those with cognitive needs who may use a screen reader to convey the image's meaning to them.

Alternative text, or "alt text" for short, is a textual substitute for non-text content in web pages, mobile applications, and documents. Effective alt text is meaningful and descriptive of the image, animation, or video, and its subject and actions.

If you don't provide content for the alt text tag in HTML, the screen reader will announce either the image filename (a brutal series of numbers and letters perhaps!) or nothing – only "image." This only tells the user they've encountered an image on the screen.

Here are some considerations when it comes to alt text:

- Use alternative text to describe meaningful images concisely.
- Provide detailed descriptions for complex images via captions or linked text.
- Ensure text alternatives convey the same information as the image.
- Ask the developer to mark decorative images properly to avoid unnecessary distractions for screen reader users. This means that you won't add a text alternative as it's not necessary.
- If text on the page provides enough information without the image, mark the image as decorative. But if the image has its own meaning, add alt text.

> There's a misconception that all images need a description, but that is the strength of a content author to say when it is and isn't necessary. Decorative images don't need alt text, if they're just providing a visual for presentation style. The WAI alt Decision Tree is a popular tool that content authors use. It provides a series of yes/no questions to help you decide whether alt text is needed for an image: `https://www.w3.org/WAI/tutorials/images/decision-tree/`.

It's also worth considering graphs, charts, and icons in the context of alt text.

Graphs and charts

Graphs and charts fall under complex images and so need alt text. But here are some specific considerations:

- Provide text-based summaries of key insights and data relationships. This assists screen reader users who may not be able to make interpretations and analyses just based on the diagrams.
- Label axes and data points so it is clear what the diagram is showing. Ultimately, this benefits everyone!
- Offer alternative data views – for example, the same information, but in a table. As discussed, people with cognitive needs sometimes prefer a data table with clearly defined relationships across a recognized block of information. This can be easier to follow, rather than multiple lines and color-coded visuals on a chart.

Icons

Icons are uniquely visual and can offer your target audience instant recognition of your chosen subject, if conventional ones are used. For example, the help icon is often a question mark, while the search icon is often a magnifying glass.

What's interesting is that icons stand alone without a visible content label a lot of the time. That's by design, but the most inclusive approach would be to combine an icon with a text label that can be read by a screen reader.

Now, do keep in mind that the label should describe the icon's function, not a description of the icon itself. As mentioned, if you use the magnifying glass icon, the alt should be "Search" (not "Magnifying glass").

Here are some more universal icons and the recommended alt text:

- Magnifying glass icon: "Search" for a search input field
- Shopping cart icon: "Shopping cart" or "View cart" in a transaction flow
- House icon: "Home" for a link or button that returns the user to the homepage
- Envelope icon: "Email address" or "Contact us" for a link or button that opens an email agent or directs the user to a web page with a contact form
- Phone icon: "Call us" next to a link of a phone number that launches a mobile phone call
- Printer icon: "Print this page" on a button that launches the native print function in a browser
- Download arrow icon: "Download [file name]" as a button, or next to a link, to launch a download on desktop or mobile
- Trash can icon: "Delete" or "Remove" on a button or next to a link that launches this function
- User profile icon: "Profile" or "Account settings" on a button or next to a link that opens a new page or popup with profile information and/or account settings
- Gear or wrench icon: "Settings" on a button or next to a link that launches a page or popup to change settings within the website or web application, or evoke native app settings.

Include text labels alongside icons when you can, and encourage the visual designer to leave room for this on the page or screen. This effort supports all users, but especially people with low vision, learning difficulties, and neurodivergence.

> Sometimes, designers will place an icon in front of a text link to make it stand out on the page, or provide a clear visual indicator. A good example is an email icon next to an email link. As long as the link text is descriptive and meaningful of its purpose or destination, there's no need to add alt text to the icon. It can be defined as decorative only.

One exception to this is the icon that designers will place before or after a link to indicate that the link opens the page in a new browser window or tab. This icon is an informative image, not decorative. The icon needs alt text so non-sighted users are aware that the link opens a new window. Use `alt="opens new window"` to do this. Alternatively, the image can be decorative, and the link itself can have an `aria-label` attribute with the visible text. For example, if the visible label is "Learn more", the code could be `aria-label="Learn more about Mortgages. Opens new window."`

Avoiding text in images

We've looked at the importance of alt text, but let's consider another image-text issue: text in an image cannot be parsed by screen reader software, so it's completely lost to non-sighted users.

For example, banner ads are bad design. They contain a bunch of text that is often crammed into a restricted layout, making the font size small and often hard to see, especially by people with low vision. For all content on the page, mobile screen, or document, work with the UI designer and inform them to avoid using images with text embedded inside.

Instead, use real text on the page, and work with the UI designer to create a background image underneath. For the text, ask them to use systemic fonts such as Arial or Verdana for the most readability.

> For practical reasons, sometimes it's just not realistic to follow this guidance. Use system fonts whenever possible, but in cases where the image has text in it, that text needs to be reflected within the alt value.

Format combinations for images

If you use images, ensure you have your alternative in place for improved accessibility:

- If you use an image, make sure you also provide alt text or ensure decorative images are appropriately hidden from assistive technologies.
- If you use an icon, make sure it is accompanied by alt text or a visible label. If you use a non-decorative, standalone icon, make sure it has appropriate alt text. Or use an adjacent visible label (like a link) and treat the icon as decorative.
- If you use a complex image (pie, bar chart, diagram, or infographic with text and images), pull out the text and format it in a plain language description. Place this text adjacent to the infographic.
- If you use a data table, consider sharing the same information in a plain language paragraph.

As a wrap-up, remember that images can be powerful, as long as they are combined with descriptions for those with visual impairments. Using a mix of images and text can also benefit people with learning disabilities or neurodivergence. They are not bombarded with a wall of text or too many visual images without clear information in the text.

Video and audio

When it comes to video and audio content – essentially anything that is primarily a visual, moving image that captures a story – here are some general best practices:

- Provide captions and transcripts for spoken content
- Offer audio descriptions for visual elements
- Use clear audio and avoid background noise
- Ensure high contrast in visuals
- Ensure the multimedia controls are keyboard accessible

These practices support users who are hard of hearing or have deafness, blindness, low vision, memory loss, learning difficulties, or are neurodivergent (ADHD, autism).

To dig a little deeper, let's focus on two points mentioned – captions and transcripts.

Closed captions

Captions are text overlays on a video that display what is being conveyed by the audio – for example, spoken dialogue and important sounds on- and off-screen. They are the most critical media alternative for video content as they are a replacement for audio.

Two types of captions are open and closed. Open captions are always visible on the video, while closed captions can be turned on or off by the viewer in the video player. For accessibility, it's preferable to use closed captions for this reason, and you should always have them. However, it's important to use text titles in the video, especially to introduce the subject at the beginning and have the name and title of a speaker onscreen.

Here are the best practices to optimize the impact of captions for people who are deaf or hard of hearing. Similarly, people who are temporarily unable to hear sound will also benefit:

- Captions should match spoken dialogue and sound effects precisely.
- Captions should clearly indicate who is speaking, especially in group conversations. Use full names in the first instance, and then reference that person by first name if there is limited room onscreen.
- Captions should include relevant non-speech audio (e.g., [applause], [door creaks]).
- Keep captions concise, limit text per line (preferably 32 characters max), and break at natural speech pauses.
- Provide translations if needed to improve accessibility.
- If you use a software platform or YouTube Studio to manage your video content and captions, you will control this. Sync captions with speech in the audio and allow enough time for users to read them.
- Software platforms can auto-generate captions for you, and YouTube Studio has this as a free feature. However, the accuracy is said to be between 70% and 80%. Always proofread and edit them for clarity and correctness.

Text transcripts

Whereas captions appear on the video screen, transcripts are written versions of the content provided as a separate text document. The written text includes what's in the captions – descriptions of the audio, spoken dialogue, and important sounds – but it should also include descriptions of what happens on the screen from beginning to end, so no information conveyed visually in the video is missed.

Like captions, transcripts provide a media alternative to an audio track to people who are deaf and/or hard of hearing. They support some users with cognitive needs, memory loss, or attention and focus deficits (ADHD) who prefer reading text at their own pace instead of watching or listening to an entire video at once.

Here are some best practices for creating text transcripts:

- Transcribe spoken words verbatim and accurately represent the speaker's tone and intent.
- Ensure you describe in detail the opening scene, titles, and text that appears, then provide a description of the speakers, objects, and actions. If needed, include any actions or sounds that take place off-screen that are important to the story.
- As with captions, preface the dialogue with the full name of the person who is speaking. You can refer to their first name only for the rest of the transcript if needed.
- Include relevant sound effects, music cues, and important background noises (e.g., [laughter], [phone ringing]).
- Break text into paragraphs, use timestamps (if needed), and maintain readability.
- Ensure correct grammar, punctuation, and clarity before publishing.
- Provide a text-based format (e.g., `.txt`, `.doc`, or HTML) for easy indexing and accessibility.
- Provide downloadable versions (e.g., plain text, PDF) for flexibility.

Sign language

According to University of California research (`https://mayberrylab.ucsd.edu/papers/Chamberlain%26Mayberry08.pdf`), sign language – not written English – is the first language for many Deaf users. Reading captions means instantaneously code-switching into a second language with a different syntax and word order. In an online learning experiment with 62 Deaf sign-first adults by Yoon & Kim in 2011 (`https://www.researchgate.net/publication/51669191`), captions boosted comprehension but did not reduce mental effort. Sign-language videos were still preferred.

So, if possible, consider offering sign language in your pre-recorded video. Search for an agency in your region to provide sign language interpreters (there are over 300 global sign languages). They record a signing session that can be embedded into the video output.

Format combinations for video and audio

If you choose the video, animation, or film format, ensure you have your alternative in place:

- If you use video, include closed captions that can be turned on and off and controlled in the video player. Include a text transcript with audio descriptions written into the text.
- Create a separate video with an audio description track embedded.

- If you use an audio clip, include a text transcript.
- If you use an animation that plays for longer than three seconds, include a text transcript describing the purpose, actions, and any sound. Flashing elements that repeatedly flash for longer than 3 seconds can induce seizures due to photosensitivity.
- If you use an animated GIF, as with static images, include meaningful alt text.

To wrap up, video and audio can be powerful formats for content, but these formats exclude people with certain needs and disabilities. So, remember to always provide a text alternative in the form of captions, a transcript, and titles for the video and audio files.

Social media

Social media refers to audio, video, text, images, icons, illustrations, and any other media formats that people use to communicate and create communities online. The intent is to share an opinion or idea, or promote your story, message, or information. Social media platforms to post your content include YouTube, TikTok, Facebook, X (formerly Twitter), LinkedIn, Snapchat, and Reddit.

As a content creator, here are some accessibility best practices for preparing social media content:

- When writing text posts, capitalize each word in multi-word hashtags (e.g., #AccessibleDesign instead of #accessibledesign).
- Whenever possible, provide descriptive alt text to non-text content such as images and GIFs. Some CMS platforms don't have this built in – speak to someone about updating this.
- Avoid excessive use of emojis or special characters when writing a post. This is because emojis come with alt text, which can lead to inaccuracies. For example, the emoji of two hands pressed together is often used to express thanks, but is actually illustrating a prayer – this could lead to a situation where your written context is saying "thank you" but the emoji's alt text says "pray." Also, emoji alt text can be very literal. For example, the alt text for the eggplant emoji would be "eggplant," but this emoji is used to denote something quite different.
- Keep your posted messages concise and structured.
- Provide video captions and transcripts.

Again, apply the principle of providing a content alternative for different users. These best practices support users with blindness, low vision, learning difficulties, and neurodivergence (ADHD, autism), and those who are deaf and hard of hearing.

As a wrap-up, apply the principle of keeping multiple users in mind when designing online content postings. Remember how different users will experience your message. Providing a variety of content formats shows your consideration for their needs.

Next, let's look at two case studies for applying best content practices.

Quick case studies

Here are two scenario examples about approaching text and images. Both explore the content author's perspective, but also provide the user's perspective as they experience the content.

Case study 1: Sarah, a content author working with infographics

Content author's perspective

Sarah, a content creator, is designing an infographic about climate change impacts. The UI designer she's working with has supplied some graphics (icons of people and clouds, pie charts and statistics, and arrows that show the cycle of precipitation in weather). Sarah wants to ensure that non-sighted users can access the same information as sighted users.

Here are some challenges she could face:

- Infographics are highly visual, relying on charts, images, and icons
- Blind users rely on screen readers, which cannot interpret visual elements
- Important data could be lost without proper accessibility measures

Now let's see how Sarah could make the infographic more accessible:

- Sarah could write **detailed alt text** that summarizes the infographic's key points – for example:

  "This infographic explains five major impacts of climate change: rising temperatures, melting ice caps, extreme weather, sea level rise, and biodiversity loss. Each section includes a statistic and a relevant icon."

- Below the infographic, Sarah could also provide a **text version** of all the content. In this case, she could write the following:

  "Title: Climate Change Impacts

  1. Rising Temperatures – Global temperatures have increased by 1.2°C since pre-industrial times.

  2. Melting Ice Caps – Arctic ice has shrunk by 13% per decade.

3. Extreme Weather – More frequent hurricanes, droughts, and wildfires.

4. Sea Level Rise – Coastal cities are at risk due to rising ocean levels.

5. Biodiversity Loss – Many species are endangered due to habitat destruction."

- Sarah could also provide a **downloadable plain text or structured PDF version** of the infographic for users who prefer alternative formats.

User's perspective

In this case, the user is blind and using a screen reader to navigate the web page. As they navigate the page with the screen reader speech output on, they use the down arrow key and land on the infographic (a `.png` file).

Firstly, the screen reader announces "image" and then reads aloud the short, detailed alt text that Sarah created and the developer coded into the image. This gives the blind user an indication of what they encountered (image) and a quick summary of the purpose of the content. Next, the user presses the down arrow key to continue navigating the page. They land on a collapsible content section (accordion), which has a heading that Sarah provided ("Climate change impacts") and functions with a button to expand and collapse it. The screen reader announces "Button. Climate change impacts. Collapsed."

By following these steps, Sarah ensures that blind users can access the infographic's information too. This improves inclusivity, user experience, and compliance with accessibility standards.

Case study 2: Kavana, labelling buttons on a website

Content author's perspective

Kavana is a content author for a retail company's website. The company has several job openings, so they want to add a call-to-action button on the homepage to drive users to the landing page of their job site. Kavana has been reading the book *Don't Make Me Think* by Steve Krug. It provides an illustration with three button examples, each with a different label. There is a scale from "Obvious" to "Requires Thought" across the top. A man, representing the website user, is looking at each button and trying to make a decision about which one to select.

On the "Obvious" end of the scale, the button label says "Jobs." The second button label is in the middle, "Employment Opportunities". The third button label is at the "Requires Thought" end of the scale. It says "Job-o-rama." Kavana wants to make an inclusive choice about the wording for the label. In the end, she goes with "Jobs."

User's perspective

Kavana starts to think about how the user will understand and interpret the label she chooses.

"Jobs" is the simplest and clearest descriptive label for the button/link and destination. The label "Employment Opportunities" is a bit long for a button label design and takes up space space on this particular page. The user will need to take more milliseconds to calculate the meaning and conclude that this button must also lead them to a jobs page.

With the "Job-o-rama" label, the user is forced to figure out the purpose or destination of the button. It could be jobs, or something more. What is an "o-rama"? What does this refer to? It may not be understood by non-native English speakers. The user will ask themselves, will the button take me to a job listings web page, or is that an event called Job-o-rama? Way too much time will be spent guessing before the user can act.

These two examples give insight into the decision-making process by both the content author and the end user. In the final section, I will share an exercise you can do to gather case studies and scenarios. This will strengthen your consistent, inclusive writing practice.

Conducting the Needs Walkthrough exercise

The Needs Walkthrough exercise is an activity you can do at any time to test-drive your content idea or prototype against the needs of five different disability types. The goal is to think through the needs of users with disabilities and ensure you're meeting them with your content. This gives you more knowledge and confidence that you're being accessible and inclusive.

The exercise itself is totally low-fi and usually doesn't cost a thing, except your time. You can run the exercise alone or with your team just by reviewing content together and asking, does this meet people's needs? If you wanted to engage real users directly, any costs associated would be compensating people with disabilities for their time in preliminary user interviews.

Let's take a look at how to prepare and complete the exercise.

Step 1: Gather a list of disability needs

If you can, gather user experiences and disability needs directly from people with disabilities. Send out a survey using Google Forms or have people answer questions on a Word document. Or, arrange one-on-one or group user interviews through accessible video conferencing platforms such as Zoom, Google Meet, or Microsoft Teams. These platforms have built-in, live captions that display for folks who are hard of hearing.

If opting for the interview approach, you have two options:

- **Option 1 – Gather general user needs only**: Keep the interview open-ended and conversational. Host the interview by asking someone about what types of assistive technologies they use, how and when they use them, what designs work best for them, and which designs cause barriers. Fable Tech Labs connects businesses with a community of people who use assistive technologies and test their products for them. Fable provides two helpful articles: *Planning Your User Interview* (`https://makeitfable.com/article/planning-your-user-interview/`) and *5 Questions You Should Ask In Your Interviews* (`https://makeitfable.com/tutorial/5-questions-you-should-ask/`). After a video interview, you can review a recording of the session and text transcript to reference later.
- **Option 2 – Use the interview to conduct the walkthrough exercise**: A second option is to gather general user needs from people during the video interview, but also use this time to have them walk through your user journey/prototype with you. Get instant feedback on the good and bad. See the full details about the exercise in *Steps 2 to 4*.

If you can't connect with real users, the *W3C Diverse Barriers and Abilities* web page (`https://www.w3.org/WAI/people-use-web/abilities-barriers/`) is a great alternative. It offers a list of the five disability types and their needs and preferences when using digital products.

After reviewing these, read *Stories of Web Users* (`https://www.w3.org/WAI/people-use-web/user-stories/`) and leverage that as a starting point for the exercise. Create a spreadsheet or checklist. Copy and paste the person's general bio and list what design works for the user, and then note the designs that cause barriers for them. This way, you can build out a library of personas to use.

Step 2: Identify the standards and outcomes to test

You now have an understanding of user experiences – what works and doesn't work within design. Now it's time to gather the standards you want to test when you run the Needs Walkthrough exercise. Usability is essential, but it's also good to test your design prototypes to see if they meet legal accessibility standards at the same time.

Walk through your content using WCAG

The Web Content Accessibility Guidelines (WCAG) is the global standard for digital accessibility and are referenced by laws in many countries. As mentioned previously, WCAG is broken down into four categories: Perceivable, Operable, Understandable, and Robust (POUR).

You can look at the entire list of guidelines on *How to Meet Quick Reference Guide* `https://www.w3.org/WAI/WCAG22/quickref/`. On this page, you can filter the list by content. This will show the guidelines (success criteria) that are related to your work.

The W3C has also created two helpful supplemental guides for meeting the WCAG success criteria for cognitive and learning disabilities (`https://www.w3.org/WAI/WCAG2/supplemental/#cognitiveaccessibilityguidance`) and low vision (`https://www.w3.org/WAI/WCAG2/supplemental/#-low-vision-accessibility-guidance`).

Test against functional requirements

EN 301 549 is an accessibility standard in Europe. Its list of functional requirements are simple user statements for meeting disability needs and it's highly recommended that you add them to your checklist/spreadsheet. Simply cut and paste them into your spreadsheet. The functional requirements are that, for any information communication technology, the following points are possible:

- Usage without vision
- Usage with limited vision
- Usage without hearing
- Usage with limited hearing
- Usage with no or limited vocal capability
- Usage with limited manipulation or strength
- Usage with limited reach
- Minimize photosensitive seizure triggers
- Usage with limited cognition, language, or learning

The Government of Canada website has the functional requirements listed here, along with the other requirements related to product design, both physical and digital (`https://accessible.canada.ca/en-301-549-accessibility-requirements-ict-products-and-services-1`).

Step 3: Plan your exercise event

Now you have both usability and accessibility testing measurements to test your design idea or prototype, it's time to plan the event, your Needs Walkthrough Exercise.

Ideally, you'll gather UI designers, UX designers, and developers to review the design prototypes together, not just from the written content lens but the entire design of the user interface experience. You can review the user stories, and then the standards guidelines and functional requirements together.

However, if you want to run the Needs Walkthrough Exercise on your own, you can do so and learn a lot about how people with disabilities interact with just your content. Perhaps you only have your ideas in a document, for the written text, labels, instructions, and alt text choices.

Step 4: Conduct the walkthrough exercise

Now it's time to conduct the exercise. Let's see what actions to take.

Action 1: Review the user stories and standards

In a team or alone, go into your checklist/spreadsheet and familiarize yourself with the user stories that represent up to five disability types, gathered from user interviews or the WAI resource. Keep in mind what they said works and doesn't work for content.

Next, review any checklists you've created and keep those in mind as guidelines that support the user stories.

Remember to keep in mind which assistive technology is being used (if any) during the user journey. For example, if you're walking through the experience as a blind user using keyboard and screen reader software, think about how the speech output will sound, especially if the headings on the page are out of order or when links are used.

Action 2: Replicate each of the five user journeys

Replicate the five disability types, like so:

1. Considering the first disability type, use your prototype and walk through your content idea or design mock-up, capturing the user journey or task from end to end.
2. Repeat this exercise for the other four disability types.
3. If you have time, consider combining disability types (for example, imagine you are using your prototype as someone with ADHD and low vision).
4. With the needs and standard guidelines in mind, assess the user journey against your research and plan. Did you come up against a barrier for the user? If so, proceed to *Action 3*.

Action 3: Record any barriers you faced

As the user, something doesn't feel right, or you've been completely stopped in your tracks by the design and cannot proceed. Ask yourself these questions and make notes in the spreadsheet:

- What is the nature of the barrier? Is the content unclear, or is there an alternative missing on a content format type?
- What is the severity of the barrier faced?
    - **Critical**: Users with this disability cannot use any part of a system, website, app, social media post, and so on
    - **High**: Users with this disability will find it challenging to use the system, website, app, social media post, and so on
    - **Medium**: Users can proceed with limited functionality
- Did the barrier fail a WCAG success criterion? If so, which?
- Did the barrier fail a 301 549 functional requirement? If so, which one?
- What is your recommended fix?
- (If relevant) Will you keep the content format but ensure there's an alternative?
- (If relevant) Would it be easier to manage, and better for end users, to convey the information in the simplest way possible?

Action 4: Record your solutions

You've now captured low, medium, or severe accessibility and usability barriers with the prototype that you can fix or remove at this early stage. Using your knowledge gathered from the user stories or interviews and standards, leverage these resources to make suggestions about content changes. Record them in your spreadsheet.

Step 5: Share what you've learned

Now you've completed the exercise for up to five disability types, and possibly a few combined types, here are some ways to share what you've learned:

- Record any additional user journeys that came about as a result of walking through your prototype that may be specific to your company's product or tasks. Write them out in the same way as the W3C *Stories of Web Users*, including a name, the disability, assistive technology used, design and experience preferences, and barriers to avoid.

- Extract all your notes from your original spreadsheet, along with any user interview recordings, videos, and transcripts – and share them in a central hub with your team. Review them regularly to keep knowledge fresh.
- Replicate the user journey on a live site or mobile app. It never hurts to unplug the keyboard and/or run a screen reader over a web page or document to experience it yourself!

Step 6: Identify who solves the issues

Later on, an additional step is to identify who is responsible for fixing the issues and barriers you uncovered. You've likely recognized the barriers caused by content as your own issues to tackle; however, sometimes you will need to work with a designer or developer to remove a barrier completely.

The W3C has also published a helpful **Accessibility Roles and Responsibilities Model** (**ARRM**) resource called *Tasks Involved in Accessibility* to complement the **success criteria** (**SC**) that make up the WCAG guidelines. Each criterion has a primary, secondary, and contributor owner – the person responsible for ensuring that the criterion is met. There are ones for content authors, UX designers, UI/visual designers and frontend developers.

This is another great team exercise to review what needs to be fixed, how, and by whom, as a team. Visit the ARRM website to see which WCAG SC are primarily owned by content authors: `https://www.w3.org/WAI/planning/arrm/tasks/`.

As you can see, the Needs Walkthrough exercise is highly customizable as a practice, with the goal of getting you to walk through a user experience from start to finish, as someone with each disability type. Doing so allows you to quickly recognize barriers, in order to rewrite content, re-consider the format that's been chosen, or add alternatives if needed. You can do this alone as a content author, applying your knowledge of user needs to your own text, images, and multimedia content, or you can conduct the exercise together with visual and user experience designers and developers to come up with solutions to all aspects of the project.

Summary

In this chapter, you have learned that it's essential to keep multiple users in mind when creating content. You've learned about the power to communicate inclusively and reach as many end users with plain, simple language for your message. This is whether it's on a web or mobile screen, in a document or on a social media platform. You've learned the importance of easily understood labels and instructions for buttons and controls. Truly, content is king, but now we have knowledge of different people's needs, you know why clarity is also king.

When applying your content strategy, you've learned to choose the most inclusive content format that meets the needs of most people with disabilities. If you select a format that excludes certain users, such as images, videos, and audio clips, it's okay to use them. You just need to provide a text alternative for people with visual and auditory impairments.

You have developed the skill of thinking inclusively as you create content. However, you want to ensure you're continuing to capture and meet the disability needs of your audience. To do this, you learned how to conduct a free and effective exercise you can do to test-drive your written content and format choices against the needs of different people. You can do the exercise alone or in collaboration with designers and developers on your team. People are at the center of your strategy. You are playing your part in changing the digital world.

In the next chapter, you will look at how to develop accessible websites, particularly focusing on semantic HTML, CSS, and ARIA.

7

Developing Accessible Websites: HTML, CSS, and ARIA Best Practices

By Crystal Scott, CPWA

Have you been a web developer for a substantial amount of time but are new – or relatively new – to accessibility? If so, I'd like you to pause and think about something: Why are you only now learning about accessibility? Why wasn't it covered when you first dove into web development? It's a sobering question, isn't it?

Accessibility is becoming a cornerstone of modern web development, yet it often isn't taught comprehensively in boot camps, degree programs, or even on-the-job training. Think about it: Did your program teach you that contrast is crucial for creating visually effective designs? Why didn't they teach you about WCAG contrast ratio guidelines while they were at it? They could have – and they should have – yet it was excluded.

Why is that? The reasons vary: tight curricula, a focus on "job-ready skills," and even a lack of awareness about the legal and ethical implications of neglecting accessibility. Or, perhaps the faculty does not have the knowledge they need to include accessibility. Unfortunately, this leaves many developers starting their careers with a glaring knowledge gap.

It's an omission that speaks volumes. Not including accessibility doesn't just hurt users; it can also create legal and financial consequences. Approximately 4,500 web accessibility lawsuits were filed in state and federal courts in 2023 against companies for failing to meet accessibility standards. The web isn't an optional resource – it's a necessity. Treating accessibility as an afterthought is a huge design flaw.

Now imagine if accessibility principles were taught alongside your first lessons on HTML, CSS, or JavaScript. How different would your understanding of the web – and its users – be today? The goal of this chapter is to help you adopt an accessibility-first mindset, a way of thinking that integrates accessibility into every decision you make.

In this chapter, we will cover the following topics:

- Structuring content with semantic HTML
- Styling for accessibility with CSS
- Enhancing dynamics with ARIA
- A developer's responsibility to test

My sincere hope is that by the end of this chapter, you'll never be able to unlearn what you've learned about accessibility. You may still encounter moments when something you build isn't 100% accessible, but I hope you'll know it. And I hope it will bother you, not out of guilt, but because you'll understand what that means for someone who is trying to use your creation. The web is for everyone, so let's start coding like it.

> An accessibility-first mindset means putting inclusivity at the core of every decision. It shifts the focus from designing for an assumed "default user" to designing for everyone.
>
> Ask yourself: Who am I creating this for? What is the intended purpose and outcome? Have I considered how this feature will work for users relying on a screen reader, a keyboard, or voice control? These questions lead to better, more thoughtful designs that work for all users.
>
> Neglecting accessibility isn't just a missed opportunity – it excludes people from life-changing products and services. By making inclusion a priority, you're building a fair, equitable, and welcoming web.

Structuring content with semantic HTML

Can you remember the first **Hypertext Markup Language** (**HTML**) project you ever created? My money is on the infamous "Hello World" project. Ah, the beginner coder days – learning to wield the mighty < and > symbols for opening and closing tags, and figuring out how a simple / turns an opening tag into a closing one. Those moments of discovery were magical.

I still remember something from my first *Intro to Programming* class through The Last Mile that has stuck with me all these years. The instructor explained HTML, CSS, and JavaScript using the analogy of the human body. HTML was the bone structure – the foundation. CSS was the skin, hair, and eye color, providing the styling on top of the skeleton. And JavaScript? That was the brain – the control center that brought functionality to the entire body. Just like a body needs strong bones, a website needs well-structured HTML for stability.

Understanding how HTML provides structure is only the beginning. To truly build accessible, meaningful websites, we need to go deeper, starting with the difference between non-semantic and semantic HTML.

Defining non-semantic HTML

To appreciate semantic HTML, it's important to first understand its opposite, **non-semantic HTML**. This refers to generic elements such as `<div>` and `<span>` that structure a web page without conveying meaning or context.

As the developer and author Manuel Matuzović says, "A div is not a button." The `<div>` element is a "pure" container. According to MDN Web Docs: "As a 'pure' container, the `<div>` element does not inherently represent anything. Instead, it's used to group content so it can be easily styled using the class or id attributes, marking a section of a document as being written in a different language (using the lang attribute), and so on."

Figure 7.1: Reminder: A <div> is not a <button>! #A11y (A11y = Accessibility)

The problem with non-semantic HTML is that it provides structure but no meaning. It's like putting content inside an invisible box – screen readers and search engines can't tell what it's for, or sometimes even that it's there. The `<div>` and `<span>` elements, for instance, don't communicate the purpose of their content, making it harder for assistive technologies and search engine crawlers to understand or interpret the page correctly.

Defining semantic HTML

Semantic HTML structures a web page using elements that inherently describe their purpose, making content more meaningful and accessible. Tags such as `<header>`, `<article>`, and `<section>` provide clear definitions of their role, allowing both browsers and assistive technologies to interpret and navigate the page effectively. This improves readability, enhances accessibility, and strengthens search engine optimization by giving content a logical hierarchy.

By using elements according to their intended role rather than visually styling non-semantic elements, developers create a well-structured and intuitive web experience. For example, `<footer>` naturally indicates closing information, while `<nav>` clearly defines a group of navigational links. These distinctions provide clarity, ensuring that users, especially those relying on assistive technology, can interact with web content in a meaningful way.

Let's take at this code block:

```html
<!-- Semantic HTML: Native button -->
<button type="submit">Submit</button>

<!-- Non-semantic imitation: Recreating a button with a <div> -->
<div
    role="button"
    tabindex="0"
    onclick="submitForm()"
    onkeydown="if (
        event.key === 'Enter' || event.key === ' ') submitForm();"
>
    Submit
</div>
```

The first example uses a semantic `<button>` element, which is accessible by default and supports keyboard interaction, screen readers, and form submission without extra code. The second example shows a `<div>` with `role="button"`, `tabindex="0"`, and JavaScript event handlers added to imitate button behavior. This approach increases complexity and risks missing critical accessibility features.

Why does semantic HTML matter?

Semantic HTML enhances your website through the following:

- **Meaningful markup**: Built-in roles (e.g., `<header>` as a "banner") and elements such as `<nav>` and `<main>` provide meaningful structure for screen readers, enabling easier navigation. By using semantic HTML, developers create a web experience where users who rely on assistive technologies can efficiently locate content, bypass repetitive sections, and navigate the page just as seamlessly as those using a mouse or touch.
- **Logical hierarchy**: Proper use of semantic headings (`<h1>` to `<h6>`) ensures content is structured intuitively, improving navigation for all users, including those using assistive technologies. In addition to aiding screen readers, a well-defined heading structure also enhances the visual hierarchy, guiding users' eyes naturally through the content. Since most people scan web pages rather than reading word for word, clear headings make it easier to locate key information quickly.
- **SEO**: Tags such as `<article>` and `<h1>` improve content discoverability and search rankings. Just like assistive technologies, search engines rely on semantic HTML to understand the structure and relevance of content, helping them index pages more accurately and rank them effectively.
- **Assistive technology support**: Elements such as `<form>` and `<label>` work together to create accessible workflows, ensuring users relying on assistive technologies can navigate forms effectively. The `<label>` element provides clear text descriptions for form fields, while `<fieldset>` and `<legend>` offer context for grouped inputs. Proper semantic markup ensures users can complete forms smoothly using a keyboard or voice commands, making interactions more inclusive and accessible.
- **Keyboard navigation**: Interactive elements such as `<button>` support keyboard use and communicate states effectively. Unlike `<div>` or `<span>`, which require additional scripting to be interactive, native HTML elements such as `<button>`, `<a>`, and `<input>` are automatically focusable and respond to keyboard events, such as `Tab`, `Enter`, and `Space`. Using semantic elements ensures that interactive components are accessible out of the box, without needing to script fixes for event handlers or keyboard focus.

Using proper semantic markup creates a roadmap for assistive technologies, enabling users with disabilities to navigate more effectively. Beyond accessibility, semantic HTML improves collaboration and SEO. Developers can understand your code more easily, and search engines can better categorize your site. Mastering semantic HTML means writing empathetic, logical code that benefits developers, users, and machines alike.

Semantic HTML lays the structural foundation for accessibility by giving content meaning and clarity. But structure alone isn't enough; how we *style* that content also plays a crucial role in usability and inclusion. Let's turn our attention to CSS and explore how thoughtful styling choices can support a more accessible experience for all users.

Styling for inclusive design with CSS

Cascading Style Sheets (**CSS**) is the stylist of modern web design, transforming basic HTML structures into visually engaging and responsive experiences. But the way you implement CSS can make or break accessibility.

Accessibility in CSS isn't just about looking good – it's about *working for everyone*. Poorly considered styling can create barriers for users with low vision, cognitive challenges, or motion sensitivities. A high-contrast button could mean the difference between completing a purchase or leaving a website frustrated. A keyboard user relies on visible focus styles to navigate, while someone with motion sensitivity might need animations turned off.

The good news? By adopting thoughtful best practices, you can ensure your CSS-enriched designs are inclusive, usable, and still beautiful.

Ensure sufficient color contrast

If users can't read your content, no amount of clever design will save it. **Web Content Accessibility Guidelines** (**WCAG**) set minimum contrast ratios to ensure legibility:

- **Normal text** (such as body copy) must have a contrast ratio of 4.5:1 against its background
- **Large text** (18pt regular or 14pt bold) requires a lower minimum ratio of 3:1 because its size makes it easier to read
- **Meaningful user interface components** (such as form field borders or focus indicators) and graphical objects (such as icons or charts) also need a contrast ratio of 3:1 with their background to remain clear and distinguishable

Let's take a look at this example:

Figure 7.2: Comparing poor and sufficient color contrast for decorative text

The top example shows white cursive text on a light purple background, which fails WCAG contrast requirements and is difficult to read for many users. The bottom example uses dark purple text on the same background, passing contrast checks and improving readability while maintaining the visual style.

Failing to meet these standards can make content difficult or impossible to perceive for users with low vision, color blindness, or those affected by glare or poor lighting conditions. Now imagine a web page using cursive low-contrast text that might look "stylish," but if users can't read your content, then what's the point? By ensuring proper color contrast, you're handing users a bright flashlight, not leaving them to fumble in the dark.

Avoid adding important content with CSS

CSS pseudo-elements (`::before` and `::after`) are useful for decorative purposes but shouldn't be used to insert critical content. Screen readers may not interpret pseudo-element content consistently; it might be ignored, duplicated, or read out of context.

If you're using pseudo-elements to add labels, instructions, or required field indicators, there's a high risk that users relying on assistive technologies will miss them. Pseudo-elements such as `::before` and `::after` don't exist in the DOM, so you can't apply ARIA attributes to them directly. For example, if you use a decorative icon with `::before`, a screen reader might still announce the generated content – unless the parent element is wrapped or styled in a way that includes `aria-hidden="true"` on an actual DOM element.

A more accessible approach is to add decorative content within a real element, like this:

```
HTML
<button>
    Visible text
    <span aria-hidden="true" class="icon"></span>
</button>
CSS
icon::before {
    content: "*"; /* Replace with an actual icon or use a background
                  image */
    display: inline-block;
    margin-left: 0.5rem;
    font-size: 1rem;
    color: #666;
}
```

Always include essential content directly in the HTML, inside the DOM, using meaningful, semantic elements such as `<label>`, `<p>`, or other tags that communicate structure and intent. Avoid generating meaningful content with CSS pseudo-elements, as their accessibility support is inconsistent. If content needs to be read by a screen reader or interacted with by users, it should be part of the actual DOM where it can be reliably announced and styled appropriately. In addition to ensuring content is accessible and meaningful in the HTML, developers should also respect system-level user preferences that affect how users perceive and interact with the interface.

Respect user preferences

Some users rely on operating system or browser settings to reduce visual strain, motion sickness, or light sensitivity. These preferences can include reduced motion, dark mode, increased contrast, or larger default fonts. When developers respect these settings, they create experiences that adapt gracefully to individual needs, without requiring users to take additional steps.

For example, the `prefers-reduced-motion` media query allows you to detect when a user has requested minimal motion and adjust animations or transitions accordingly:

```
CSS
@media (prefers-reduced-motion: reduce) {
    * {
        animation: none;
        transition: none;
    }
}
```

Another common preference is dark mode, which can help reduce eye strain, improve readability in low-light environments, and support users who are sensitive to bright screens or have photophobia. You can detect and respond to this setting using the `prefers-color-scheme` media query:

```
CSS
@media (prefers-color-scheme: dark) {
    body {
        background-color: #121212;
        color: #f5f5f5;
    }
}
```

By honoring system-level preferences like these, you're giving users control over how they consume content, supporting comfort, clarity, and accessibility without forcing visual experiences that may cause discomfort. Respecting user preferences is just one part of building adaptable interfaces. Developers also need to ensure that designs remain usable and accessible across a wide range of screen sizes, zoom levels, and device configurations.

Design for flexibility and scale

Responsive design is about more than screen size. It also ensures accessibility across devices, zoom levels, and user settings. According to WCAG 2.2 Success Criterion 1.4.10: Reflow, content must remain usable without requiring users to scroll in two directions at once. Specifically, users should not have to scroll horizontally when the viewport is reduced to 320 CSS pixels wide or 256 CSS pixels tall. A common way to test this is by zooming a page to 400% on a 1280-pixel-wide screen and/or using **Device Mode** in browser developer tools. This ensures that content reflows properly and remains functional for users who rely on screen magnification or small displays.

To meet these needs, use relative units and flexible layouts that adapt gracefully:

- Use `rem` for font sizes, which scale relative to the root (`<html>`) element, and respect user-defined preferences.
- Use `em` for spacing within components, allowing padding and margins to adjust based on text size.
- Avoid fixed pixel (px) values for text and layout, as they don't respond to browser zoom or OS-level scaling.
- Ensure layout reflows at narrow viewports without cutting off or hiding content. Use techniques such as `flex-wrap`, media queries, or CSS grid to allow elements to stack or shift naturally:
    - `flex-wrap`: Allows flex items to wrap onto multiple lines, preventing them from shrinking too small or overflowing their container.
    - **Media queries**: Apply different styles at various breakpoints, enabling layouts to adapt responsively across screen sizes.
    - **CSS grid**: A powerful layout system that lets you create two-dimensional layouts, adjusting columns and rows to fit different screen sizes.
- Test all interactive elements – including forms, buttons, and links – to confirm functionality and visibility at high zoom and on small screens.

By combining flexible CSS with relative units, you can create a responsive, accessible experience that works well for everyone, including users with low vision or those navigating on mobile devices.

Provide highly visible focus styles

Focus indicators are visual cues that highlight which interactive element is currently focused, such as a button, link, or form field. These indicators typically appear when users navigate a page using a keyboard (e.g., with the *Tab* key), helping them keep track of where they are on the page.

To apply focus styles effectively, it's important to understand the difference between `focus` versus `focus-visible`:

- `*:focus`: This applies focus styles whenever the element is focused, including when clicked with a mouse or navigated to via a keyboard
- `*:focus-visible`: This specifically targets keyboard users (or input methods that rely on focus navigation) and does not apply styles when the element is clicked with a mouse

Some developers remove focus styles entirely because they conflict with their visual design, often without realizing how essential these indicators are for accessibility. Without them, keyboard users are left guessing where their focus is on the page, making navigation confusing and frustrating.

Here's an example of what not to do:

```
*:focus {
    outline: none;
}
```

This completely removes the browser's default focus styles, leaving keyboard users without any visual indication of where they are on the page.

Instead, enhance focus indicators to align with your site's design while keeping them clearly visible:

```
*:focus-visible {
    outline-color: #212121;
    outline-style: auto;
    outline-width: 2px;
    outline-offset: 2px;
}
```

The `outline-style: auto` keyword allows the browser to apply its default outline appearance, which may vary depending on the platform or user agent. While this can help maintain consistency with native focus styles, it is important to ensure that your outline has sufficient contrast and visibility against your site's background.

The `outline-offset: 2px` CSS property moves the outline slightly outside the element's edge, preventing it from being visually compressed against borders or overlapping with shadows; this improves clarity and helps the outline maintain a visible 3:1 contrast ratio on both light and dark backgrounds.

Ensuring visible focus is just one way CSS supports accessibility. Next, let's look at why inclusive design choices in CSS matter for real users.

How CSS supports inclusive design

Inclusive design ensures that visual styling works for everyone, not just those using a mouse, a default font size, or a bright screen. When implemented with care, CSS helps remove barriers, reinforces content structure, and allows interfaces to adapt to a wide range of user needs, devices, and preferences. Here are a few of the outcomes inclusive design supports when CSS is used thoughtfully:

- **Better readability**: High-contrast text and visible focus indicators help users with low vision or color vision deficiencies (color blindness) perceive and interact with content comfortably
- **Clear separation of content and decoration**: Keeping meaningful content in HTML and using CSS for styling only ensures that screen readers interpret content correctly
- **Reduced motion sensitivity**: Honoring user preferences to reduce motion prevents discomfort and distraction for users affected by motion-triggered conditions
- **Responsive scaling**: Using relative units such as `rem` and `em` allows layouts and text to scale naturally, supporting zoom and custom font sizes without breaking the design
- **Flexible layouts**: Responsive CSS adapts your design across screen sizes and zoom levels, ensuring content stays functional at 400% zoom or on narrow viewports
- **Keyboard-friendly navigation**: Highly visible focus indicators help keyboard users understand where they are on the page and navigate efficiently

Even with well-structured HTML and accessible CSS, there are cases where native elements alone cannot provide all the information or interaction needed. This is where ARIA comes in. When used correctly, ARIA can enhance accessibility by filling in gaps and making dynamic content and custom components more understandable to assistive technologies.

Enhancing dynamics with ARIA

Accessible Rich Internet Applications (ARIA) is the toolset developers turn to when native HTML falls short. Introduced by the **World Wide Web Consortium** (**W3C**), ARIA bridges the gap between dynamic web components – such as modals, accordions, and sliders – and accessibility. By using ARIA attributes programmatically, developers can help assistive technologies interpret context and behavior in situations where native HTML alone is not sufficient.

Before ARIA, developers struggled to build interactive components that were understood by assistive technologies. And let's be honest, many still do – which is exactly why guidance like this exists. ARIA attributes give developers a way to programmatically communicate the purpose, state, and relationships of custom components when native HTML falls short:

- `aria-expanded` can tell assistive technologies whether an element – such as a drop-down panel, accordion section, or menu – is currently expanded or collapsed.
- `aria-labelledby` is commonly used to label elements such as accordion panels or tab panels by referencing the button or tab that triggers them. This ensures that assistive technologies announce a clear and descriptive label based on visible content.

It's important to remember, though, that ARIA is like salt. When used at the appropriate time and in the right amount, it can make food taste incredible. But, if used at the wrong time or in the wrong amount, it can ruin the experience.

The five rules of ARIA

The W3C outlines these essential principles for using ARIA effectively:

1. **Use native HTML elements first**: If you can use a native HTML element or attribute with the semantics and behavior you require already built in (instead of re-purposing an element and adding an ARIA role, state or property to make it accessible), then do so.
2. **Do not change native semantics, unless you really have to**: Don't override the meaning of HTML elements by changing their roles. For example, don't assign `role="tab"` directly to a heading like `<h2>` – wrap it properly instead.
3. **All interactive ARIA controls must be usable with a keyboard**: ARIA does not provide keyboard functionality. If you build a custom widget with ARIA roles (like `role="button"`), you must script keyboard behavior manually – including focus, Enter, and Space keys.
4. **Do not use `role="presentation"` or `aria-hidden="true"` on focusable elements**: These attributes remove elements from the accessibility tree. If the element is still focusable, it can confuse screen reader users by making focus land on "nothing."
5. **All interactive elements must have an accessible name**: Any interactive element (like a button, link, form input, or custom widget) must expose an accessible name to assistive technologies. This helps users understand the purpose of the control.

The first rule is critical: use native HTML elements first. Before you reach for ARIA, take a close look at what semantic HTML can already do. It's the quiet overachiever of web accessibility – built-in elements come with meaningful roles, behaviors, and keyboard support that sometimes even ARIA can't replicate.

If a native element exists, use it. ARIA is a powerful tool, but it should be your last resort – not your first instinct. Nothing says "I didn't do my research" like slapping `role="button"` on a `<div>` when you could've just used `<button>`.

Before applying ARIA customizations, always refer to the latest specifications to ensure you are using roles, states, and properties correctly. Guessing or misapplying ARIA can lead to confusion for assistive technologies, ultimately harming rather than helping accessibility. For example, adding `role="navigation"` to `<button>` instead of using a proper `<nav>` element may confuse screen reader users by misrepresenting the element's function and purpose.

The latest ARIA specification, WAI-ARIA 1.2, became a W3C Recommendation on June 6, 2023. It provides detailed guidance for using roles, states, and properties effectively: `https://www.w3.org/TR/wai-aria-1.2/`. For further best practices, consult the Using ARIA guidelines, published as a W3C Working Draft on September 27, 2018: `https://www.w3.org/TR/using-aria/`.

Now let's take a closer look at the two ARIA features we mentioned at the start of the section: `aria-expanded` and `aria-labelledby`.

aria-expanded

The `aria-expanded` attribute communicates whether content associated with an element is currently displayed (`true`) or hidden (`false`). Dropdowns, accordions, and disclosure panels are common use cases for this attribute. Non-sighted screen reader users especially benefit from its presence, as they cannot rely on visual cues such as arrow icons or sliding animations to understand whether content has been revealed.

Consider a common FAQ interaction, where a user clicks a question to reveal the answer. Visually, the change is obvious. But for screen reader users, we need to communicate whether the associated content is expanded or collapsed. The `aria-expanded` attribute makes that possible.

```
HTML
<button aria-expanded="false" class="faq-toggle">
    What is your return policy?</button>
<div class="faq-answer hidden">
```

```
    <p>You can return items within 30 days of purchase for a full
refund.</p>
</div>
CSS
hidden {
    display: none;
}
JavaScript
const button = document.querySelector('.faq-toggle');
const answer = button.nextElementSibling;

button.addEventListener('click', () => {
    const expanded = button.getAttribute('aria-expanded') === 'true';
    button.setAttribute('aria-expanded', String(!expanded));
    answer.classList.toggle('hidden');
});
```

This setup uses `aria-expanded` to communicate the state of the answer content, giving screen reader users the same context that sighted users get from visual cues. The `.hidden` class ensures the content is visually hidden without relying on the HTML `hidden` attribute, which behaves differently and can cause issues with toggling in JavaScript.

aria-labelledby

`aria-labelledby` is used to programmatically associate a visible text label with another element on the page. This is especially useful when the text label is not nested inside the element being labeled but is visually associated with it. A common example is labeling a form section using a heading that describes its purpose.

Imagine a checkout page where the payment form is visually introduced by a heading. To ensure screen readers announce the form with a meaningful name, the developer can link the form container to the heading using `aria-labelledby`:

```
<h2 id="payment-heading">Payment Information</h2>
<form aria-labelledby="payment-heading">
    <!-- Form fields here -->
</form>
```

These are just two examples of the many ways ARIA can be applied to enhance accessibility. When used thoughtfully, attributes such as `aria-expanded` and `aria-labelledby` help communicate structure, relationships, and behavior to assistive technologies.

Applying ARIA to enhance dynamic and custom components

Mastering ARIA means knowing its purpose and limits. ARIA isn't a substitute for semantic HTML – it's a supplement for when native elements fall short. Here's how to make the most of ARIA:

- **Know when to use it**: Follow the five rules of ARIA to ensure you're enhancing accessibility, not hindering it.
- **Check the specs**: Always consult the latest ARIA guidelines to apply roles, states, and properties correctly.
- **Communicate changes**: Use ARIA attributes such as `aria-expanded` and `aria-labelledby` to communicate states and relationships between elements. These attributes – and others like them – help assistive technologies interpret dynamic behavior and provide meaningful labels for interactive controls.
- **Test for compatibility**: Validate ARIA implementations with assistive technologies to confirm they work across devices and browsers.

By thoughtfully applying ARIA, you can ensure that custom components are accessible, intuitive, and inclusive for all users. However, writing accessible code is only part of the process; you also need to verify that it works as intended. The next step is testing your code to catch accessibility issues early and ensure your implementation holds up in real-world scenarios.

Testing your code for accessibility conformance

One of the most essential best practices in *developing accessible websites* is learning how to test your own code. This chapter has focused on HTML, CSS, and ARIA, but even the most carefully written code won't result in an accessible experience unless it is tested. Accessibility is not just about writing perfect code. It is about achieving conformance with the WCAG and delivering the best possible user experience for the widest possible audience. And that starts with knowing how to test your work in real-world conditions.

Developers must understand the guidelines they are designing and coding for. Testing should not be left solely to auditors or quality assurance specialists. If you're building digital products, testing for accessibility conformance should be part of your development workflow from the beginning, not a final step. This means understanding how to identify common barriers, interpret WCAG requirements, and recognize when something that *looks* fine may not be usable for everyone.

A foundational accessibility testing process should include both automated tools and manual testing techniques. Automated tools are helpful for catching clear-cut errors, such as missing alternative text or insufficient color contrast. However, many accessibility issues, such as incorrect focus behavior, mouse-only interactions, or misleading link text, can only be caught by testing manually.

The manual process must include the basic use of screen reader technologies. You don't need to be a screen reader expert, but you should be familiar with how screen readers work, how to navigate a site with one, and how your content is announced to screen reader users. Whether you're using NVDA on Windows, VoiceOver on macOS or iOS, or another tool, this knowledge helps you catch critical usability issues that automated scans will miss.

It is also important to test your websites and components on multiple devices. User experiences vary across operating systems, browsers, screen sizes, and input methods. A design that works well on a desktop browser may break or become unusable when zoomed on a mobile screen or viewed in high contrast mode on a tablet. Testing across real environments ensures your work is inclusive by design – not by assumption.

Browser developer tools play a key role in this process. The accessibility tree, focus order inspection, and role and name checks available in Chrome DevTools and Firefox are indispensable for understanding how assistive technologies interpret your code.

All together, you should build up an accessibility testing toolkit – this is not just a collection of tools but a core part of how you work as a developer. The goal is not to use everything available but to consistently use the tools that align with your workflow, support your responsibilities, and help you build for real users.

Every developer should maintain a toolkit that includes both automated and manual testing methods. This means checking for issues with the accessibility tree, validating semantic structure, inspecting color contrast, and simulating user experiences with assistive technologies. Testing should be an integrated part of development, not a final task or something handed off to someone else.

Here are the tools I use most often in my own accessibility-focused development work:

- **Chrome DevTools**, especially the **Accessibility** pane and accessibility tree inspection
- **Webflow Audit panel**, for identifying common accessibility issues during the build process
- **TPGi Color Contrast Analyzer**, which I use to test contrast between foregrounds and backgrounds
- **ARC Toolkit**, for validating landmarks, roles, and structural markup
- **WAVE browser extension**, for quick, visual accessibility feedback on web pages
- **NVDA** for screen reader testing on Windows
- **VoiceOver** for macOS and iOS
- **TalkBack** for testing on Android devices
- **A reliable pixel-to-rem converter**, to help ensure scalable, user-friendly sizing

This is a short list based on what works for me. You may find other tools that better suit your platform, role, or workflow. What matters most is consistency and intention. Testing should happen throughout development, not as an afterthought.

As your projects evolve, so should your toolkit. Make time to learn new tools, explore how users experience your site, and adapt your process to meet the needs of more people.

While this chapter introduces the importance of testing, *Chapter 11* will go much deeper. It will walk you through specific tools, methodologies, and workflows to help you build a sustainable, repeatable approach to evaluation.

For now, the most important takeaway is this: testing is part of development. It is not optional, and it cannot be fully outsourced. Developers who take ownership of accessibility by learning the guidelines, practicing manual testing, using assistive technologies, and evaluating their work on multiple devices are the ones building experiences that are truly inclusive, usable, understandable, and welcoming to the widest possible audience.

Summary

This chapter explored the practical foundations of accessible web development, covering semantic HTML, inclusive use of CSS, appropriate application of ARIA, and the importance of accessibility testing. You have learned how HTML provides structure and meaning, how CSS can support usability without introducing barriers, and how ARIA can extend accessibility when native HTML is not enough. You also saw why testing your work for accessibility should be part of every developer's process, and how a thoughtful testing toolkit can help you identify and fix barriers early.

These skills are essential in today's development landscape, where the digital experience must serve a broad range of users with diverse needs. Whether you're building a landing page, a web app, or a complex interactive component, accessibility isn't a nice-to-have; it's a non-negotiable part of quality code. Adopting an accessibility-first mindset empowers you to create products that are more usable for all, legally compliant, and human-centered by design.

In the next chapter, you will explore how to extend inclusive design principles to mobile environments, where touch gestures, screen size, and voice interaction introduce a unique set of accessibility challenges and opportunities.

Further reading

Accessibility is not a one-time fix. It is an ongoing commitment that requires continuous learning and adaptation. As standards, tools, and technologies evolve, staying current helps ensure your work remains effective, inclusive, and aligned with real user needs. Make time to review updates to WCAG and WAI-ARIA, participate in accessibility communities such as WebAIM and the W3C, and follow trusted blogs, newsletters, and webinars to keep your knowledge sharp. Beyond implementation, advocacy also plays a critical role. Whether you are mentoring peers, publishing insights, or leading conversations within your team, your influence can help embed inclusive practices across your organization and the broader industry.

Here are a few trusted resources to support your continued growth:

- **W3C Web Accessibility Initiative** (`https://www.w3.org/WAI/`): The W3C's official hub for accessibility standards, including WCAG and WAI-ARIA. This is the go-to source for technical specifications, guidance, and working group updates on digital accessibility.
- **WebAIM** (`https://webaim.org/`): A widely respected organization offering practical accessibility articles, research, and training. WebAIM is especially helpful for beginners and teams looking to build foundational knowledge.
- **Deque University** (`https://dequeuniversity.com/`): A robust online training platform with courses on accessibility testing, screen reader usage, and ARIA. Deque's resources are widely used in both enterprise and educational settings.
- **Accessibility Book Club** (`https://www.a11ybookclub.com/`): A free, community-driven group I founded for developers, designers, and accessibility advocates who want to learn together through monthly readings, discussions, and live events.

By staying connected to these resources, you will continue growing as an accessibility-minded developer and contribute to a more inclusive web. Because if your website isn't working for everyone, it's not really working at all.

8

Mobile Accessibility: Designing for Touch and Voice Interfaces

By Chris McMeeking

What are you doing when you use your phone? Are you at work? Sitting in the passenger seat of your car? Using a wheelchair? Taking a jog? Mobile devices are used by all sorts of people in all sorts of contexts. Bringing the convenience of mobile apps to your users in all these contexts is what accessible mobile application design and development is all about. However, there are many challenges in your way. In some ways, mobile accessibility mirrors the concepts of web accessibility, though there are a few unique challenges that must be considered, including:

- Small screen sizes built for interaction with universally clumsy fingers.
- Expectations of support for platform features that differ between mobile platforms (i.e. iOS or Android).
- The need to split complex experiences into workflows that may be interrupted by users on the go.

These are among many others that will be discussed over the course of this chapter.

In this chapter, mobile designers will discover how to craft inclusive, intuitive interfaces, while developers will learn how to implement those designs using platform-specific accessibility tools. Together, you'll create mobile experiences that welcome every user, on every device, through every input method. By understanding the differences between iOS and Android platforms, optimizing touch targets, implementing alternative input methods, designing effective voice interfaces, and ensuring responsive layouts, you will gain practical strategies to make your mobile applications more inclusive.

As mobile devices remain the primary way people engage with digital content – accounting for over 60% of global web traffic as of 2024, according to WebAIM Million Report (2024) and StatCounter Global Stats – designing with accessibility in mind enables you to reach a broader audience, improve user experiences, and build applications that stay resilient as platforms evolve.

So, in this chapter, we will cover the following topics:

- Mobile development accessibility considerations
- Optimizing touch targets for various abilities
- Implementing alternative input methods
- Designing voice interfaces for accessibility
- Ensuring responsive layouts across devices

Mobile development accessibility considerations

When developing accessible mobile applications, it's crucial to recognize the differences between iOS and Android platforms. While both operating systems offer robust accessibility frameworks, their features, conventions, and limitations can vary significantly. Understanding these distinctions helps ensure your applications provide a seamless experience for all users, regardless of the platform they're using. The following section compares accessibility considerations specific to iOS and Android.

Regarding iOS vs Android

An important thing to remember in mobile accessibility is that iOS and Android accessibility are as different as web and mobile accessibility. There are many unifying factors, but there are also many factors unique to each platform.

Much of the differences come from the nature of the languages and ecosystems upon which the platforms are built. The web is built on top of a series of technologies and specifications created by committee, and evolves rather slowly. Meanwhile, Android and iOS are built by largely singular entities driven to evolve their platforms on a yearly basis. The faster Apple and Google evolve their respective platforms, the more users are driven to buy new devices.

This rapid platform evolution represents challenges to accessibility. Users expect support for new platform features, but when these features are not supported, applications feel unwieldy.

Let's take a look at the ecosystems these applications are built in, starting with Android:

- **Operating System**: Android
- **Programming Language**: Java or Kotlin
- **UI Framework**: Compose or XML Layout

And now let's look at iOS:

- **Operating System**: iOS
- **Programming Language**: Swift or Objective-C
- **UI Framework**: SwiftUI or Storyboard

Note the level of control Google and Apple leverage over their respective ecosystems. Google works with the open source community, though still exercises a lot of power and final review over their contributions, while Apple owns its ecosystem in its entirety, including the programming language.

Because Google and Apple have such control, a recurring theme across this chapter will be doing things the way the ecosystem supports them. It is important to give the platforms the space to be different. Without these differences, you will be forcing one of the platforms to behave in a way that is contrary to its supported features. This has a compounding effect of making your application more difficult to maintain, therefore falling behind on platform updates and exacerbating these problems.

> When the distinction between iOS and Android solutions is important, it will be mentioned and discussed separately.

Regarding maintainability

One of the most critical – and often overlooked – distinctions in mobile accessibility is the difference between web content accessed through a mobile browser and native mobile applications. While they may look similar on the surface, their underlying technologies, capabilities, and accessibility frameworks differ significantly.

Mobile browsers often support a subset of desktop web standards, giving the illusion that web accessibility practices translate directly to native apps. However, native applications are built using entirely different architectures, with platform-specific controls, compiled languages, and stricter privacy rules between components. This creates unique accessibility challenges that don't arise on the web. Additionally, hybrid and multi-platform frameworks – like React Native, Flutter, and Xamarin – further blur these lines. While they offer the convenience of shared codebases, they also introduce complexity when aligning accessibility expectations with platform-specific behaviors.

Recognizing these distinctions is just as important as understanding the differences between iOS and Android, and it forms the foundation for making informed, accessible design decisions on mobile. This leads to conversations between accessibility experts and developers that can end up with the developer implementing custom functionality where it is not needed in order to support functionality that is not expected.

Forcing mobile applications to behave like their web counterparts creates problems for the user and causes applications to diverge from the behaviors and implementations that Google and Apple expect. When a new version of Android or iOS is released – which happens frequently – existing controls may not immediately support new assistive technology features introduced with the update. Accessibility experts should understand the intended behavior of controls within the platform's evolving framework to ensure compatibility and accessibility.

In an ever-changing ecosystem, this can be a challenge. Some ways to combat this are to:

- Compare your application to Google or Apple applications while accepting that they may not conform to your expectations coming from a web application.
- Thoroughly familiarize yourself with multiple platforms' assistive technologies before sharing comments on your preferred testing choice.
- Be open-minded in discussions with engineers, especially when they say something can't be done.

These strategies can help you navigate the complexities of mobile accessibility, but it's essential to remember that the underlying architecture of mobile applications presents unique challenges. Unlike web development, mobile apps are built using compiled languages, which enforce strict privacy rules between different components of the application. Keeping this in mind is crucial as you approach accessibility improvements.

Mobile applications are built on compiled languages. Compiled languages have strict privacy expectations between various bits of implementation. Every time we ask developers to add custom functionality to improve accessibility, we are actually asking them to violate this contract. While in certain situations it may be possible, it will lead to maintainability issues and may in fact diverge from the expected behavior of the control.

Allow the platform to function as it's designed to, even if its behavior differs from what you're accustomed to in a web ecosystem. Both iOS and Android offer robust, built-in accessibility support, but custom implementations that override default behavior can inhibit that support. Asking developers to work around perceived platform weaknesses often leads to applications that are harder to maintain and less accessible over time. When platforms eventually address those weaknesses, custom solutions may conflict with the new expected behavior, unintentionally diminishing accessibility instead of enhancing it.

Incorporating platform conventions is essential for building accessible, maintainable applications. By respecting native behaviors and avoiding unnecessary customizations, you can ensure compatibility with evolving accessibility standards. With a solid foundation in place, the next step is to optimize touch targets to accommodate a diverse range of user abilities.

Optimizing touch targets for various abilities

Mobile devices present unique accessibility challenges because users often interact with them while on the move, using their fingers to tap, swipe, and pinch. This can be particularly difficult when:

- **Controls are too small**: If a control is smaller than the fingertip, users may struggle to tap it accurately, especially when moving or holding the device with one hand.
- **Controls are crowded together**: When multiple controls are placed close to each other, it can be hard to determine which control is being activated, leading to errors and frustration.

These challenges can be even more pronounced for users with mobility issues, such as those with limited dexterity, tremors, or other motor impairments. Designing controls with adequate size, spacing, and touch responsiveness is essential to create a more inclusive mobile experience.

So, how do we account for touch target size without letting interactive controls dominate the user interface? While not limited to mobile, WCAG 2.5.8 Target Size (Minimum) (`https://www.w3.org/WAI/WCAG22/Understanding/target-size-minimum`) addresses this issue across all touch interfaces – including tablets, laptops, and assistive technologies used by people with limited dexterity – by recommending a minimum target size for interactive controls. The current recommendations are a minimum size of 24 x 24 CSS pixels for AA conformance with a recommended AAA threshold of 44 x 44 CSS pixels. Let's also look at some more helpful concepts.

The card pattern

The card pattern is a design approach that organizes information into distinct, interactive containers or "cards." These cards are typically large enough to be easily tapped or clicked, making them particularly accessible for mobile users, including those with motor impairments or limited dexterity.

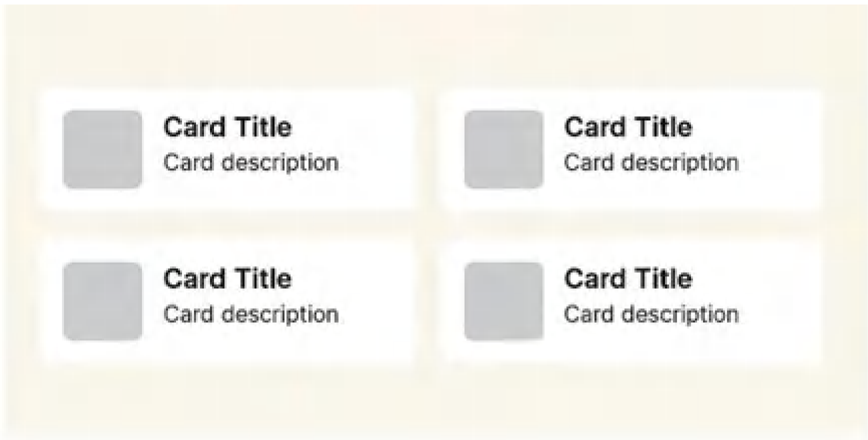


Figure 8.1: The card pattern

By grouping related content and interactive elements within clearly defined cards, you can enhance usability and accessibility. These are especially effective when presenting lists of items, such as products, articles, or app listings. They allow for generous tap targets and visual separation between different pieces of content, reducing the risk of accidental interactions.

For practical examples of the card pattern, visit the Google Play Store or the Apple App Store. Notice how apps, games, and other content are presented as distinct cards, each occupying enough space to be easily tapped. This design ensures clarity, consistency, and accessibility.

However, it's important not to overload cards with too much information or nest multiple interactive elements in conflicting ways. For example, assigning an entire card to navigate to a user's profile, while also placing a tappable email address or button inside it, can lead to confusion or unintentional interactions. To ensure a clear and accessible experience, keep each card focused, avoid overlapping interactive zones, and prioritize a single, predictable action per card when possible.

So, the card pattern is one effective way to combine information and interactivity in a space-efficient layout. It brings key content to the forefront while maximizing the touchable area for interaction. While cards are especially well-suited for mobile, they're not the only way to group related elements. What matters most is ensuring clarity, focus, and generous touch targets, regardless of the visual pattern you use.

Two-dimensional scrolling

Two-dimensional scrolling is a design technique where users can navigate content both horizontally and vertically within the same interface. This pattern is commonly used on mobile devices to allow for efficient browsing of large amounts of content without overwhelming the user. For example, users can scroll vertically through a feed or a list and horizontally through categories or image galleries.

The infamous carousel pattern – a form of horizontal scrolling often implemented as a series of swipeable cards or images – is frequently criticized in web design for several reasons:

- **Poor accessibility**: Carousels often rely on animations or gestures that may be difficult for users with motor impairments or those who navigate using keyboards or screen readers.
- **Content visibility**: Important content can be hidden off-screen, making it less discoverable. Many users never interact with carousel controls, leading to missed information.
- **Overuse and misuse**: Carousels are often employed for aesthetic purposes without considering usability or accessibility, resulting in a poor user experience.

However, on mobile devices, two-dimensional scrolling and carousels are essential. They allow users to quickly navigate large datasets or categories, particularly in applications like Netflix or online stores where content is organized into horizontal rows of cards. Without carousels, finding a specific movie or product would require endless vertical scrolling.

Two-dimensional scrolling also pairs well with the card pattern, enabling designers to maximize touchable areas for controls while presenting content in a visually organized and accessible way. However, it's important to ensure that horizontal scrolling is used only for discrete content groups – not for reading or core navigation – to meet WCAG 2.1 Success Criterion 1.4.10 Reflow, which requires that content can be presented without loss of information or functionality when resized or zoomed.

While two-dimensional scrolling can enhance navigation and efficiency, it's not always the right choice. It can introduce cognitive overload if too many scrollable sections compete for attention or if users lose track of their position. Designers should avoid nesting multiple scrollable areas and ensure that horizontal scrolling is clearly indicated and easy to perform. When overused or poorly implemented, this pattern can hinder discoverability and frustrate users, especially those relying on screen readers or Switch Access. Use it thoughtfully, and always test it with real users.

Floating action buttons

One challenge with mobile accessibility is balancing large touch targets with the limited screen space available for displaying content. To maximize usable space, developers often use fixed-location floating action buttons – interactive controls that appear to hover above other UI elements. This technique allows the content beneath to scroll independently, providing a larger, more accessible touch target without sacrificing information visibility.

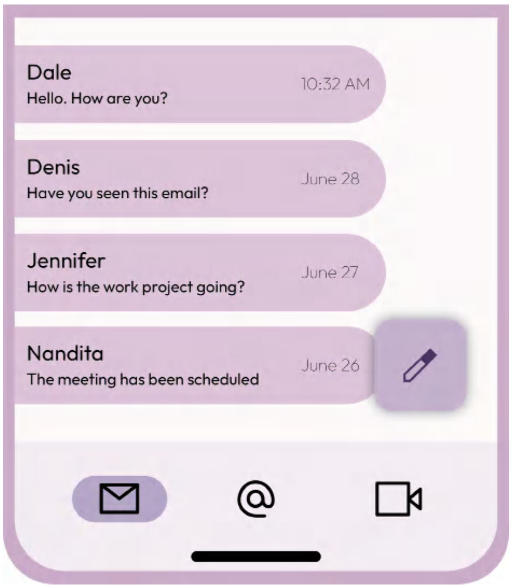


Figure 8.2: Floating action button

In web design, floating action buttons are often criticized for poor discoverability and confusing focus order. When users navigate with keyboards or assistive technologies, it can be difficult to determine where floating action buttons fit within the sequence of interactive elements. This can compromise conformance with WCAG 2.4.3 Focus Order and 1.3.2 Meaningful Sequence, both of which require a logical and predictable navigation path. Additionally, floating action buttons may obscure important content, leading to usability and accessibility issues, especially for users relying on screen readers or Switch Access.

However, in mobile design, floating action buttons are widely accepted and effective for several reasons:

- **Platform standards**: Both iOS and Android provide established patterns for implementing floating action buttons. These guidelines include recommendations for size, placement, interaction, and accessibility. Following platform conventions ensures consistent user experiences.
- **Simplicity of layout**: Mobile interfaces typically contain fewer interactive elements than web pages, reducing the likelihood of confusion in focus order. Additionally, developers can leverage built-in platform tools such as Android's ViewGroup attributes (android:nextFocusForward, etc.) and iOS's UIAccessibilityContainer protocol to manage and customize focus order in an accessible way.
- **User expectation**: Many users have come to expect floating action buttons in mobile apps for primary actions – like composing a message or adding a new item – because of their widespread use. However, this familiar pattern may not be equally intuitive or accessible to everyone, particularly users who rely on screen readers, Switch Access, or keyboard navigation. When considering affordance, it's important to ask: who benefits from this pattern, and who might be excluded by it? Designing with intention means ensuring that commonly used patterns also support users with diverse abilities.

When implementing floating action buttons, consider these best practices:

- **Placement**: Floating action buttons are generally placed in the bottom-right corner (Android) or centered at the bottom (iOS) to remain accessible while not obstructing critical content. This consistent placement also benefits non-sighted users by allowing assistive technologies – like screen readers or Switch Access – to reliably locate and describe the button's function, reducing guesswork and supporting efficient navigation.

- **Accessibility considerations**: Ensure floating action buttons are labeled appropriately for screen readers and integrated into the focus order. Typically, this means placing them near the end of the navigation sequence – after the main content – so they don't interrupt the user's reading or task flow. Avoid dynamically repositioning their focus unless context demands it (such as in a modal or step-based workflow). iOS and Android both provide accessibility APIs to help developers manage this in a predictable, user-friendly way.
- **Consistent usage**: Use floating action buttons only for high-priority actions, and avoid cluttering the interface with too many floating controls.

Extending the touchable area

Often, an interactive control – such as a button, checkbox, or input field – is placed within a larger layout container, especially in lists or forms. A simple way to improve usability and accessibility is to extend the touchable area beyond the control itself to include the surrounding layout. For example, instead of making just the small checkbox tappable, you can also make the associated text label interactive, so users can tap anywhere in that row to activate the checkbox. This is particularly helpful for users with limited dexterity or those using one-handed touch gestures. Expanding the touch target in this way improves the experience without cluttering the interface, and it's an easy win for accessibility.

Utilize workflows

Designers often struggle with limited screen space because they attempt to fit too much content or functionality into a single screen. A better approach is to break down the user experience into manageable workflows – smaller, focused interactions that guide users through a sequence of steps rather than overwhelming them with a cluttered interface.

Consider a mobile banking app that allows users to:

- Check their account balances
- Transfer money between accounts
- Pay bills
- Deposit checks

If all of these tasks were presented on a single screen, the result would be a confusing, crowded interface with tiny touch targets and poor usability. Instead, workflows can be broken down into separate, streamlined screens – in this case, four – each focusing on one task at a time:

1. **Home screen**: Shows an overview of accounts with options to navigate to specific tasks.
2. **Transfers workflow**: Guides the user through selecting accounts, entering an amount, and confirming the transfer.
3. **Bill payment workflow**: Breaks down the process into choosing a payee, entering a payment amount, and setting a schedule.
4. **Check deposit workflow**: Utilizes step-by-step guidance for photographing a check, confirming the amount, and submitting the deposit.

By breaking down complex tasks into dedicated workflows, you can improve usability, accessibility, and discoverability. Users are less likely to be overwhelmed or make errors, and each screen can be optimized with larger, more accessible touch targets.

While designing for touch accessibility is crucial, it's only one piece of the puzzle. To ensure your application is truly inclusive, it's essential to consider users who interact with their devices through alternative input methods, such as hardware keyboards, voice commands, or assistive technologies like Switch Control. But beyond input methods, thoughtful workflow design also benefits screen reader users, people with cognitive disabilities, and anyone who might feel overwhelmed by dense or complex interfaces. Supporting diverse interaction patterns and breaking down tasks into focused steps helps make your app more usable, discoverable, and welcoming for everyone. Let's explore how to design for these varied needs.

Implementing alternative input methods

Did you know that you can interact with an Android phone with a mouse? Or that there are users that use only a keyboard? You can even plug in a joystick and turn on an assistive technology like Switch Control or Switch Access!

Fortunately, when it comes to testing, you can generally narrow your efforts down to two sets of technologies:

1. Input technologies that rely on textual information from the accessibility layer (i.e. VoiceOver on iOS or TalkBack on Android)
2. Input methods that rely on detecting how it is expected to interact with a control (i.e. keyboard or Switch Access)

Supporting all of these input methods can be challenging, particularly when building custom controls. You may not always have access to the APIs needed to replicate certain behaviors, and even when APIs exist, they're often incomplete or difficult to implement consistently.

Both iOS and Android provide robust accessibility layers – like VoiceOver and TalkBack – that interact well with built-in components. These platforms also support external input devices, such as Bluetooth keyboards, switches, and adaptive controllers. Assistive technologies like Switch Control (iOS) and Switch Access (Android) can bridge the gap between software and hardware, allowing users to interact with key functionality via scanning interfaces or external switches.

However, custom controls that don't inherit from platform-native components may break this compatibility, leading to frustrating or inaccessible experiences. Whenever possible, rely on built-in features and extend from native components to ensure full support across the diverse ecosystem of inputs.

A simple design rule: platform controls or custom solutions

To create accessible, multi-input content, it's essential to follow a straightforward design rule:

- Use native platform controls whenever possible: Rely on the built-in buttons, sliders, text fields, and other UI elements provided by iOS and Android. You can use them as is, combine them to create more complex components, or build on top of them while keeping their original behavior intact. This helps ensure your app works smoothly with assistive technologies like screen readers and Switch Access.
- If custom controls are necessary, ensure they interact with assistive technologies and input devices in a way that aligns with platform expectations.

The idea is to leverage built-in platform controls (like buttons, sliders, or text inputs) or extend them using the platform's own accessibility frameworks. When you use native controls or build upon them, accessibility features – such as keyboard support, screen reader compatibility, and touch interactions – are handled by the operating system.

When you deviate from using platform-native controls and rely on custom implementations, such as custom gesture recognizers, you risk breaking compatibility with assistive technologies. These custom gestures may not be detectable by screen readers, keyboards, or other input devices, resulting in a poor experience for users who rely on them.

Developers must then put in significant effort to replicate accessibility behaviors that the operating system would otherwise provide automatically. This can lead to inconsistencies and decreased accessibility over time, especially as platforms evolve.

The Web Content Accessibility Guidelines (WCAG) recognize that some applications may need custom interactions. For example:

- Standard apps (e.g., banking app): Should adhere to native controls as much as possible to ensure accessibility across all input modalities.
- Games or creative apps: Often require custom gestures and interactions that aren't covered by typical platform controls. Accessibility may still be provided, but custom solutions are expected.

Now, in order to successfully support all of the different ways users can interact with mobile operating systems, you must stick to the things the ecosystem is capable of. This can be different between operating systems. Let's take a look at both Android and iOS.

Android keyboard support

Unlike iOS, Android keyboards interact with the operating system more like traditional desktop keyboards, complete with manageable focus navigation and tabbing behavior. This traces back to Android's early days, when devices included physical trackballs and arrow keys. Trackball navigation, keyboard support, and joystick-style inputs were all part of the original design thinking, and those foundations still influence Android's accessibility support today.

For example, consider a simple login screen with two text fields and a button: on Android, a connected keyboard (via USB-C or Bluetooth) allows users to navigate between fields using the *Tab* key and activate the button with *Enter* – behaviors supported natively by the operating system. On iOS, a Bluetooth or Lightning-connected keyboard works differently: keyboard navigation requires users to enable an assistive technology like Full Keyboard Access, Voice Control, or Switch Control to move between elements. Even then, developers must ensure focusable elements are properly exposed using accessibility APIs to make this navigation reliable.

Many applications benefit from solid keyboard support out of the box because both iOS and Android determine default focus order using depth-first tree traversal. This means the system examines the user interface hierarchy top-to-bottom and left-to-right (or right-to-left in certain locales), based on how elements are structured in the code. When developers use native components and maintain a logical view hierarchy, focus order typically "just works." But for custom interfaces or less common navigation needs, developers may need to intervene and define a more intuitive focus path.

iOS keyboard support

Supporting hardware keyboards is different between iOS and Android, as they each treat keyboards fundamentally differently. For iOS hardware, the keyboard accepts input and translates it to similar signals as the software keyboard.

What iOS does differently is that it doesn't support full keyboard focus navigation in the same way as Android by default. You can connect and use an external keyboard – via Bluetooth or Lightning – with an iPhone or iPad to type and perform some basic actions without enabling any assistive technologies. However, to navigate interface elements more comprehensively using arrow keys or tab-style navigation, users must enable an assistive technology like Full Keyboard Access, Switch Control, or Voice Control. Full Keyboard Access, in particular, unlocks more extensive navigation and interaction patterns, but it also changes how the onscreen keyboard behaves – for instance, disabling it when an external keyboard is connected.

> You only need to test with one of Switch Control, Voice Control, or Full Keyboard Access to help cover WCAG conformance from a technical standpoint, but it's important to understand that these tools do not behave identically. For example, passing tests with Full Keyboard Access does not guarantee a seamless experience for Voice Control users, since each relies on different interaction models and APIs. While it may be impractical to test every modality in every situation, being aware of their differences – and the potential for missed issues – can help you make more informed design decisions and uncover usability gaps earlier.

While iOS handles keyboard navigation differently, the concept of focus order is a critical consideration for both iOS and Android. Whether users are navigating with a keyboard, screen reader, or other assistive technology, ensuring a logical and intuitive focus order is essential for accessibility. Let's take a closer look at how mobile operating systems determine and manage focus order.

Understanding focus order in mobile accessibility

When users navigate an application using assistive technologies like screen readers or keyboards, the focus order determines the sequence in which interactive elements are accessed. Ensuring a logical focus order is critical for accessibility.

Mobile operating systems calculate focus order by performing a depth-first tree traversal of the interface hierarchy. This means the system starts at the top-most parent control and examines each child control before moving on to the next sibling. The order in which items are encountered during this traversal defines the default focus order.

Unlike web-based screen readers – or VoiceOver on macOS, which may treat nested controls as a single grouped element – mobile operating systems like iOS and Android typically treat nested controls as individual, separate elements. For example, if you have a button inside a card component, each element would receive focus independently on mobile, allowing more granular navigation but also requiring a well-structured focus order to prevent confusion:

- **Web context (VoiceOver on macOS):** May treat the card as a single grouped control depending on how the HTML and ARIA roles are implemented. If the container uses appropriate landmarks or roles and the button is properly labeled, VoiceOver may navigate to the button independently, but without a clear structure, it might only be described as part of the group.
- **Mobile context:** Both the card and the button are treated as distinct elements, with each receiving focus independently.

This difference can be beneficial for accessibility, but only if the focus order is planned correctly. A poorly defined focus order can lead to confusing and frustrating experiences for users who rely on assistive technologies.

As an engineer, the things you want to be concerned about are:

- That your view hierarchy is well structured and laid out in a sensible manner (i.e. avoid using absolute layouts and repositioning items).
- That the controls users can interact with are detected as actionable.

While the default traversal works well for many applications, thoughtful design and a well-structured UI hierarchy can often eliminate the need for manual customization. In cases where custom focus order is necessary, developers should ensure it supports a logical, intuitive experience for all users. Here are some common techniques for managing focus order effectively:

- **Explicitly setting focus order using APIs**: When the default focus traversal doesn't align with the intended user experience, developers can use platform-specific APIs to define the desired focus path. Android provides methods like `focusNext()`, `focusPrevious()`, `focusUp()`, `focusDown()`, `focusLeft()`, and `focusRight()` to explicitly guide the focus movement. iOS offers similar control through the UIAccessibility framework.

- **Defining accessibility containers (iOS)**: On iOS, developers can use the `accessibilityContainer` property to group related elements. This allows them to control how VoiceOver navigates through complex UI hierarchies, ensuring users encounter content in a logical sequence.
- **Setting accessibility focus order (Android)**: Android's View hierarchy allows developers to specify focusable elements and their traversal order using attributes like `android:nextFocusForward`, `android:nextFocusLeft`, `android:nextFocusRight`, and `android:nextFocusUp`. This is particularly useful when building custom controls or modifying default navigation patterns.
- **Programmatically controlling focus**: Developers can programmatically request focus for a particular element using functions like `requestFocus()` (Android) or `becomeFirstResponder()` (iOS). This technique is helpful when guiding users through specific workflows or highlighting important content, but should be used sparingly.
- **Leveraging focus management libraries**: In complex applications, using third-party libraries that offer focus management utilities can simplify the process. These tools often provide higher-level abstractions to control focus order, making it easier to ensure accessibility consistency across an application.
- **Testing and refining focus order**: Testing focus order with various assistive technologies (e.g., screen readers, keyboards, Switch Control) is essential to confirm that navigation is logical and intuitive. Refining the focus order based on user feedback can significantly enhance the accessibility of your application.

Usability for all users

As we have covered throughout the book, sometimes doing things for accessibility is just about doing things right. There are a few things we can consider that benefit all users, while particularly saving time for those users of alternate input devices.

Both platforms offer a feature that's often overlooked: the ability to customize the text displayed on the submit button. For example, on mobile devices, the button that usually says "Enter" or "Go" can be changed to something more specific, like "Search" or "Send," depending on the desired action. When a user is done filling out a `Text` field, there are any number of sensible things to do. Move to the next field, submit the form, "go" to a webpage, etc. Think about what users would expect to happen when they hit the *Enter* button and provide them with that contextual hint.

Also, as Android users, the back action is really important. Each Android phone supports its own way of interacting with the back action – on some devices, it's a dedicated hardware button; on others, it's a software button with a reserved place (unless it's in the context of a fullscreen app); or it can even be a gesture performed from the left edge of the screen. Regardless, consideration for what the back action should do in every context is important. Often, the default behavior will be good, but just as often you will want to customize it.

Good behavior typically means the back action matches user expectations, like navigating to the previous screen or exiting a modal. Bad behavior might involve unintentionally exiting an app, losing unsaved data, or breaking a user's flow by sending them too far back in the navigation stack.

Customizing the back action could include:

- Preventing accidental exits by prompting users to confirm before leaving a critical flow (like discarding unsaved changes in a form).
- Redirecting users to a specific screen instead of the previous one, which can be helpful for guided workflows or when maintaining a consistent navigation structure.
- Triggering specific actions within an app, such as closing a menu or collapsing UI elements instead of exiting entirely.

Tailoring the back action ensures that users have a seamless and intuitive experience, especially when the default behavior doesn't fit your app's design or user flow.

Designing for diverse input methods – whether touch, keyboard, or alternative controllers – ensures accessibility for a wide range of users. However, not all interactions rely on physical input. Voice interfaces offer another powerful way for users to navigate and control applications, providing flexibility and accessibility for those who may have difficulty with traditional input methods. Let's explore how to design voice interfaces that are both effective and inclusive.

Designing voice interfaces for accessibility

Apple and Android tend to mirror each other's operating system functionality. When a new feature appears in one, you can be sure that the other is aware and, not long later, it will often show up in its own way in the other. The same logic applies to the accessibility space.

For example, both ecosystems support the ability for users to turn on a voice-activated assistive technology. In Android, it's called Voice Access, and in iOS, it's called Voice Control. Voice Access and Voice Control work in fundamentally the same way. When activated, the user can interact with their device through structured voice commands, such as:

- **"Tap {Control Identifier}"**: This command allows users to activate a specific interactive element by referencing its accessible label (e.g., "Tap Submit" or "Tap Add to Cart"). For this to work reliably, the control must have a clear, unique, and descriptive `accessibilityLabel` (iOS) or `contentDescription` (Android). Avoid using vague or duplicate labels, as this can confuse voice interaction systems and lead to failed actions or incorrect selections.
- **"Show Names"**: This command prompts the voice control system to display the accessible names of all interactive elements onscreen. It allows users to see which elements are labeled and available for direct interaction using voice commands (e.g., "Tap Send"). If an element doesn't have an accessible name, it won't appear, which can lead to critical actions being missed. To support this feature, ensure that all actionable elements have clear, concise `accessibilityLabel` (iOS) or `contentDescription` (Android) values.
- **"Show Numbers":** This command overlays numbers on all recognized interactive elements, allowing users to interact by saying "Tap 5," for example. It's especially useful when elements don't have distinct or visible labels. However, unrecognized or improperly coded elements may not be numbered, creating barriers. Ensure that every interactive control is exposed to the accessibility layer so it can be detected and numbered when needed.
- **"Scroll {Up/Down}"**: This command enables users to navigate vertically through scrollable content, such as lists or feeds. It's especially important for interfaces with dynamic content or long pages. Ensure that scrollable areas are properly marked in the accessibility tree so assistive technologies can detect them. Poorly coded containers or hidden overflow can prevent this command from working as expected.
- **"Increment/Decrement {Control Identifier}"**: This command adjusts values for controls such as steppers, sliders, or number fields (e.g., "Increment Quantity" or "Decrement Volume"). For voice interaction to succeed, the control must be properly labeled and implemented using platform-native components or accessible custom controls that expose their roles and actions. Without this, users may be unable to change values via voice, limiting accessibility for those with motor impairments.

Designing content that works seamlessly with voice-activated assistive technologies involves ensuring your interface is accessible and responsive to voice commands. This is achieved by:

- **Using semantic HTML**: Properly structured HTML with clear headings, labels, buttons, and links makes it easier for voice technologies to interpret content accurately.
- **Providing descriptive labels and alt text**: Ensure all interactive elements (like buttons and form fields) have descriptive labels that exactly match – or at least include – the visible text label if present. This alignment helps users issue accurate voice commands (e.g., "Tap Submit") and ensures that assistive technologies can correctly associate the spoken command with the intended element. For images, use descriptive alt text or mark them as decorative to ensure accessibility for screen readers and other voice-activated tools.
- **Ensuring consistent navigation**: Predictable and clear navigation structures help voice commands work smoothly. Users should be able to reach every part of your interface using simple, consistent voice commands.
- **Testing with voice-activated tools**: Regularly testing your design with popular tools like Google Assistant, Siri, Alexa, or Dragon NaturallySpeaking ensures compatibility and usability.
- **Providing keyboard accessibility**: Since many voice-activated tools interact with content by simulating keyboard commands, ensuring full keyboard accessibility is essential.

While users can sometimes work around poorly designed content, ensuring your site follows these principles makes their experience faster, smoother, and more intuitive.

Identify controls

One consistent need of both voice-activated assistive technologies is to identify controls that users would like to interact with. This works differently between the two ecosystems:

- **iOS**: Voice Control expects very structured accurate commands with labels that identify the control precisely.
- **Android**: Voice Access accepts less precise commands and will attempt to guess which control is intended when a precise label isn't available.

Because of this, iOS has stricter expectations regarding control labeling. However, it also provides developers with tools to manage this, such as the ability to separate `accessibilityLabel` (used to identify the control) from `accessibilityValue` (used to convey its current state or content). For example, a button to apply a discount code might have an `accessibilityLabel` of "Apply Discount" and an `accessibilityValue` of "Code SAVE20 entered." This allows voice control users to say "Tap Apply Discount" while still getting feedback about the current code without confusion. Be sure to keep the `accessibilityLabel` concise and action-focused to serve as a reliable identifier.

Avoid nesting controls

Both Voice Access and Voice Control activate elements by sending a virtual tap to the center of the control. This generally works well, however, in certain circumstances, the center of the control isn't actually the correct target. If another control is nested on top of the control being activated and happens to be located at the central point, you can make it impossible for the user to activate the encompassing control.

An alternative to nesting interactive targets is utilizing custom actions. By naming actions within a component, you can avoid having several interactive targets while allowing users to use grammar like "Show Actions for {Control Identifier}." This allows you to have several actions associated with a control while avoiding the nested controls problem.

In addition to custom actions, other effective methods for designing accessible voice interfaces include:

- **Grouping related controls**: Instead of treating each control as a separate target, group related items under a single, clearly identified control. This reduces the number of interactions required and simplifies navigation.
- **Providing contextual feedback**: Ensure that users receive immediate and relevant feedback when issuing commands. This helps confirm their actions and guides them if they need to try again.
- **Ensuring consistent terminology**: Use consistent phrasing and naming conventions throughout your interface. This minimizes cognitive load and makes interactions more intuitive.
- **Supporting error recovery**: Provide clear instructions or hints when an unrecognized command is issued. Allowing users to ask for available actions (e.g., "What can I do here?") can significantly improve the experience.

These methods help create a smoother, more intuitive experience for users who rely on voice interaction.

While designing effective voice interfaces is essential, accessibility also depends on how well your application adapts to different devices and screen sizes. Mobile users engage with apps on phones, tablets, watches, and other devices, each with unique display characteristics and interaction patterns. To provide a consistent and accessible experience, it's crucial to ensure your layouts are responsive across all devices.

Ensuring responsive layouts across devices

One of the biggest challenges mobile developers face is designing for the sheer variety of devices in use, ranging from compact phones to tablets, watches, and linked external displays. Compounding this is the need to support dynamic display scaling and user-adjustable text sizes, both of which are critical for accessibility.

Supporting a broad range of devices isn't just a technical requirement – it's key to creating inclusive experiences. For example, WCAG Success Criterion 1.4.10 (Reflow) requires that content can be viewed without loss of information or functionality when resized up to 400%, which corresponds to a 320px-wide viewport in portrait orientation. Similarly, WCAG SC 1.4.4 (Resize Text) requires that users can resize text up to 200% without assistive technology and without breaking layout or usability.

According to Test Sigma, the most common mobile screen resolution in 2023 was 360×800 (11.02% market share), followed by 390×844 (7.93%) and 414×896 (5.56%). But successful design isn't about hitting specific resolutions – it's about supporting flexible, responsive layouts that adapt gracefully to any screen size or zoom level.

In spite of this, users expect support across all their devices, and you have little control over which devices your application will run on. This complexity is amplified on Android, where manufacturers like Samsung often ship their own browsers and default screen readers, potentially affecting accessibility behaviors. On Apple devices, the distinctions between iOS and iPadOS – while fewer – can still have meaningful impacts on assistive technologies and layout behavior.

Information density

We can link the concepts of device size and display scaling together by simply considering them to be different information density configurations. For example, consider a 4” device with the smallest display scale and a 6” device with the largest. These two configurations could have the same information density, even if on the 4” display, things appeared smaller – either way, the same amount of content would fit on the screen.

Given this mechanism, we can split the ecosystem into three categories of information density:

- **High**: Small 4”-5” devices with display scaling set to default or higher.
- **Medium**: 5”- 7” devices, large phones, with display scaling set lower than the default.
- **Low**: 7” devices with lower display scaling settings and larger tablet form factors.

To streamline testing without covering every screen resolution, developers can group devices by common screen characteristics. Typical categories include small phones (like older Android models or compact iPhones), standard phones (modern iPhones and Androids), large phones or phablets (Galaxy Note series, iPhone Plus/Pro Max), tablets (iPads, Galaxy Tabs), and foldables (Galaxy Z Fold, Surface Duo).

Developers should design for a representative device in each category rather than testing every device individually. For instance, an iPhone 14 Pro Max can represent large phones, while an iPad Air can represent tablets. This approach ensures efficient coverage without excessive testing.

Applying responsive design principles – like flexible layouts, scalable typography, and adaptable UI elements – helps accommodate different screen sizes and densities. Tools such as CSS media queries, viewport units, and relative sizing support this adaptability.

Testing should involve both emulators and real devices to verify that your designs respond appropriately to different screen sizes, resolutions, and scaling settings. While emulators can speed up early testing, they don’t always capture the nuances of how assistive technologies behave on physical devices, so real-device testing remains essential, especially for accessibility. To meet WCAG conformance, thorough testing across a range of input methods, screen sizes, and device types is critical. While testing based on representative categories can help streamline your process, it should never replace the diligence required to ensure inclusive, standards-conformant experiences.

Adaptive layouts

As you start testing your applications across different information densities, you will certainly come across a situation where your application doesn't function well in one. In these situations, it can be helpful to allow your application to use space differently based on the device it is installed on. There are a few common techniques for this:

- Adding a column or changing the number of columns in a multi-column layout.
- Having a portion of your application permanently visible, such as a menu on the left.
- Splitting your application into multiple panes, such as in a master detail workflow.

You can also consider custom design solutions with adaptive layouts, however, it's important to remember that your developers will need to account for things that they can get for free from platform layout containers in the above methods. This can lead to inconsistencies and bugginess in UI code that is difficult to get completely correct. This balance of achieving your design goals while utilizing platform tools is important to consider early in your design process.

Utilize standard tools

The final tip is to leverage the learnings of the platform creators. Apple and Google dominate the ecosystem because they have done their research well and know their platforms and their users. They also hold a lot of control over the future direction of their platforms and make every effort to provide backward compatibility, particularly for those aspects of their ecosystem documented in their respective design guidelines:

- **Apple**: Human Interface Guidelines (`https://developer.apple.com/design/human-interface-guidelines`)
- **Android**: Material Design Guidelines (`https://m3.material.io/`)

By leveraging the knowledge and components in these documents, you save time and maintenance pain by utilizing system components and designing applications that fit what the platform already does. Going too far outside of this bubble will leave you struggling to keep up with an ecosystem that is driven to provide users with new experiences on a yearly basis.

Summary

In this chapter, you learned how to design mobile applications that work effectively across a wide range of devices, peripherals, and user capabilities. By categorizing devices according to information density, leveraging platform-specific capabilities, and using default components, you can provide a consistent and accessible experience without the need to test for every possible device individually.

This information is valuable because it allows developers to build adaptable, resilient applications that remain compatible as platforms evolve. By aligning your designs with Android and iOS guidelines, you save development time, reduce maintenance efforts, and benefit from continuous improvements offered by the platforms. Staying within their ecosystems ensures that your application remains accessible and modernized without extra effort.

In the next chapter, you'll explore how to design accessible experiences within emerging technologies like virtual reality, augmented reality, and artificial intelligence. As these platforms grow, ensuring accessibility becomes more challenging but also more essential. You'll learn practical techniques for creating inclusive experiences, from making virtual environments navigable to ensuring AI-driven interactions are responsive to diverse needs. By applying accessibility principles to these cutting-edge technologies, you can build more inclusive and innovative applications.

Part 3

Advanced Strategies and Implementation

The final section of the book explores how to scale accessibility beyond individual interfaces and into organizational practices, team culture, and emerging technologies. These four chapters help you deepen your impact by focusing on inclusive mobile design, testing and remediation strategies, and fostering a company-wide commitment to accessibility.

You will learn how to design for mobile touch and voice interactions, evaluate accessibility using both automated tools and manual audits, and build inclusive design systems that serve a wide range of user needs. Finally, the section concludes with a roadmap for creating a sustainable accessibility culture—one grounded in empathy, shared responsibility, measurable progress, and continuous learning. Together, these chapters equip teams to move from reactive fixes to proactive, inclusive design leadership.

This part includes the following chapters:

- *Chapter 9*, *Accessibility in Emerging Technologies: VR, AR, and AI*
- *Chapter 10*, *Foundations of Accessible Design System Patterns*
- *Chapter 11*, *Tools and Techniques for Accessible Evaluation and Prioritization*
- *Chapter 12*, *Building an Inclusive Design Culture: Strategies for Organization*

9

Accessibility in Emerging Technologies: VR, AR, and AI

Dr. Keith Newton

On the day that I wrote this chapter, I found one of the most fun, free, online, time-wasting tools of all time! I logged on to `labs.google.com` and clicked on *Viola the Bird*. This AI experiment lets anyone, with or without musical expertise, play a cello-inspired instrument. Viola uses a leg to hold the instrument, another leg to balance, and grips the bow using her beak. Using a mouse, a user can control this rather enthusiastic bird to play pieces like Ode To Joy, Jupiter from Holst's The Planets, Christmas songs, and more. The music adjusts in real time to the speed and placement of the bow, as well as the timing of notes and the arrangement of the orchestration. It's super cool and super fun, but unfortunately, because of the way it was implemented, it fails a large portion of its potential users.

Don't get me wrong. I **love** *Viola the Bird*. I can imagine somewhere in an office, the creator poised in their office chair, laptop open to a cutting-edge AI prototype months in the works, and everyone on the team excited about the next "revolutionary platform for personalized content," with investors eager to see it go live. That said, the uncomfortable truth is that the prototype is navigable and operable only via mouse. Users operating a screen reader have no text alternatives provided, and the product animations don't respect user motion guidelines.

What if *Viola the Bird* had been implemented with a more inclusive mindset? Could the team have implemented keyboard navigation, implemented motion control using arrow keys in addition to the mouse, and labeled the content with text alternatives? That way, users with both visual and motion impairments would be able to navigate and operate Viola in an equitable manner that allowed them to enjoy playing those pieces and sharing in the joy of this product. The system would not be perfect, but it would be **welcoming** to everyone.

This chapter explores the accessibility challenges and opportunities presented by emerging technologies such as **Virtual Reality (VR), Augmented Reality (AR)**, and **Artificial Intelligence (AI)**, discussing the unique considerations for making immersive VR experiences accessible to users with various disabilities. The chapter places a focus on information presentation and interaction while looking at the role of AI in enhancing accessibility and potential biases.

In this chapter, we will cover the following topics:

- Adapting VR experiences for various sensory abilities
- Enhancing AR accessibility through customization
- Leveraging AI to automate the inclusion of accessibility practices
- Mitigating algorithmic bias in AI-driven interfaces

Adapting VR experiences for various sensory abilities

Virtual Reality (VR) is a technology that simulates three-dimensional, computer-generated environments and immerses users through specialized headsets or devices. By tracking head movements, and often with hand controllers, VR allows people to view content, navigate landscapes, and even manipulate virtual objects in real time.

VR has been embraced across industries: from gaming and entertainment to education and professional training, where immersive simulations of environments in fields include, but are not limited to, medicine, aviation, and industrial safety.

Yet, the very elements that make VR so engaging – things like fully 3D environments, dynamic motion, and realistic audio – can also introduce barriers for disabled users. Users with visual impairments, for example, may struggle to identify in-game elements if not provided with robust audio cues or haptic feedback. Deaf or hard-of-hearing users might miss crucial story beats or puzzle hints delivered only through dialogue or sound effects, while persons with mobility limitations may find the physical demands of holding or operating controllers difficult or impossible to manage, or find that intense motion can trigger dizziness or nausea for other vestibular disorders.

Because each design choice, from interface elements to navigation methods, can either enable or exclude certain users, it helps to acknowledge the varied abilities and needs of potential users before diving into development. It is also helpful to reflect on ways to systematically evaluate and refine the VR experience to better serve a broad range of participants.

Accommodations to improve VR accessibility

VR offers a range of customizable experiences for users, but can also present challenges for individuals with disabilities. The following list highlights benefits, challenges, and accommodations, along with examples of strategies to improve perception, operability, and understanding for all users, while offering specific adaptations for those users with mobility, visual, auditory, cognitive, and motion-related disabilities:

- **Mobility accommodations:**
    - **Challenges:** Motion controls often require specific physical positioning, and precise interactions could create barriers.
    - **Solutions:** Provide alternative input methods for all actions (voice commands, eye tracking), enable control remapping, allow input sensitivity adjustments, and include specialized tooling for personalized mobility support.
- **Visual accommodations:**
    - **Challenges:** Reading text can be difficult, color differentiation may be challenging, glasses might cause discomfort, and screen reader compatibility can be limited.
    - **Solutions:** Offer adjustable text size and contrast, haptic feedback for events, visual and non-visual labels for all content, and audio descriptions of the environment.
- **Auditory accommodations:**
    - **Challenges**: Spatial audio design may be inconsistent, directional sounds might be hard to pinpoint, and reliance on voice chat can limit accessibility.
    - **Solutions**: Incorporate mono audio options, offer customizable captions and visual cues, include signing avatars, and provide adjustable left-right audio balance.
- **Cognitive accommodations**:
    - **Challenges**: Users may experience sensory overload, disorientation, time mismanagement, and overly complex interfaces.
    - **Solutions**: Utilize teleport buttons, provide easy re-orientation options and scalable UI/HUD, offer step-by-step tutorials, and include guided interactions.

- **Motion sickness mitigation**:
    - **Challenges**: Sensory mismatches between virtual and real-world movement can cause discomfort.
    - **Solutions**: Employ higher frame rates, use teleportation locomotion instead of smooth movement, allow for user restriction of camera motion, utilize cooling fans, integrate haptic feedback, and allow gradual exposure.
- **Other considerations**:
    - **Challenges**: Flashing images may trigger seizures, XR technology can cause vertigo, and users may face difficulties readjusting after use.
    - **Solutions**: Limit flashing images, use fade transitions for teleportation, and allow stepwise adaptation within environments.

Fictional case study: virtual escape room

Consider a multinational company that decided to host its quarterly team-building exercise in a virtual escape room designed for all employees. The primary goal is to foster cross-departmental relationships by letting participants collaborate on puzzles, share insights, and celebrate small wins in real time. By engaging in this shared challenge, colleagues should build trust and rapport, leave with a stronger sense of camaraderie, and gain renewed appreciation for each participant's unique skills.

However, let's imagine that some of the users have vestibular disorders – conditions affecting balance and spatial orientation, often causing dizziness, vertigo, and disorientation. The following accommodations highlight key strategies developers can use to reduce disorientation, stabilize user perspectives, and ensure VR projects become more inclusive and ultimately more engaging for vestibular disorders:

- **Teleportation or comfort locomotion:**
    - **What it is:** Instead of moving through the environment with smooth, continuous walking or flying, users "jump" to their desired location or engage in a "low-speed" mode.
    - **Why it matters:** Sudden or continuous motion in VR can trigger intense discomfort or motion sickness. Teleportation/comfort modes reduce that discrepancy between visual and physical movement.

- **Stable reference frames:**
    - **What it is:** A virtual horizon line, cockpit, or other stationary elements that remain fixed while the user moves.
    - **Why it matters:** A consistent point of reference helps counteract disorientation, anchoring the user's balance and reducing vestibular strain.
- **Adjustable Field of View (FOV):**
    - **What it is**: Allowing users to narrow their view or enable a vignette effect focuses attention on a subject by removing distracting background detail during movement.
    - **Why it matters:** Peripheral vision can amplify motion sickness. Adjusting or reducing FOV during movement mitigates the intensity of visual input.
- **Easy access to pause and breaks:**
    - **What it is:** Clear options to pause or exit the VR environment, as well as natural breakpoints in the activity.
    - **Why it matters:** The ability to quickly step out if discomfort arises helps users manage symptoms before they escalate, encouraging continued participation.
- **Iterative user testing and feedback:**
    - **What it is:** Inviting participants with known motion sensitivities to test early prototypes, plus offering a feedback mechanism in the final experience.
    - **Why it matters:** Users with vestibular conditions can identify hidden triggers or issues more reliably than generic guidelines can, informing more targeted improvements.

This inclusive approach ultimately benefits everyone. The accommodations allow for persons with vestibular discomfort to participate equitably in an environment, and allow them to strengthen cross-departmental relationships as they are, and participate in a supportive, innovative, and cohesive organizational culture.

By prioritizing inclusivity, organizations foster an environment where all employees can fully engage and thrive. Building on this foundation, let's explore how AR can be customized to further enhance accessibility.

Enhancing AR accessibility through customization

Augmented Reality (AR) transforms how people engage with digital content by overlaying virtual elements – such as text, images, and interactive features – directly onto their surroundings. Unlike VR, which immerses individuals in a completely digital world, AR leverages cameras, sensors, and specialized software on devices like smartphones, tablets, or headsets to enhance rather than replace the physical environment.

This approach has been embraced across numerous fields: in retail, customers can virtually "try on" products; in on-site training, workers can see real-time instructions superimposed on machinery; and in educational settings, students can explore layered views of historical artifacts or scientific processes in greater depth.

A compelling illustration of AR's potential is the concept of "smart cities," where visitors use AR apps to access enhanced details about nearby locations. Users may point a phone at a building to view a 3D timeline of its restoration or scan a street sign to get additional details from the area. These features help transform passive visits into an interactive storytelling experience.

AR, similar to VR, can introduce barriers for users with disabilities. Individuals with visual impairments may struggle to see small or low-contrast text; deaf or hard-of-hearing visitors require captions or sign-language alternatives for audio-based narration; and people with mobility limitations might find it fatiguing to hold devices aloft for extended periods or perform precise gestures. Moreover, persons with cognitive limitations can become overwhelmed by the clutter of multiple on-screen layers.

These challenges underscore the importance of user-centric design. Each interface decision – whether it's text size, color contrast, or how and when digital overlays appear – can either facilitate or hinder accessibility. Testing with diverse user groups, collecting iterative feedback, and refining AR implementations accordingly, developers and institutions are able to create experiences that are more inclusive and equitable.

Designing inclusive AR experiences

AR, similar to VR, presents unique challenges for individuals with disabilities. The following list outlines challenges and solutions to improve perception, operability, and understanding in AR environments, addressing issues related to mobility, visual clarity, auditory cues, and cognitive load:

- **Mobility accommodations:**
    - **Challenges:** Interaction with AR elements may require precise gestures or specific physical positioning, which can be challenging in dynamic real-world environments.
    - **Solutions:** Provide alternative input methods (voice commands, single-touch virtual buttons that stay in one place), enable gesture sensitivity adjustments, and offer customizable control schemes to support smoother interactions.
- **Visual accommodations:**
    - **Challenges**: AR content can be difficult to read or interpret due to variable lighting conditions, background interference, and insufficient contrast between virtual elements and the physical environment.
    - **Solutions:** Offer adjustable text size, contrast, and brightness settings; implement high-contrast modes; and allow users to reposition or resize AR overlays to enhance clarity and readability.
- **Auditory accommodations:**
    - **Challenges:** Reliance on auditory cues in AR can be problematic for users with hearing impairments, and ambient noise may further hinder audio clarity.
    - **Solutions:** Incorporate customizable captions and subtitles, provide visual audio alerts, and offer options for mono audio or adjustable audio balance.
- **Cognitive accommodations:**
    - **Challenges:** The simultaneous presentation of digital and physical information may lead to cognitive overload, confusion, or distraction.
    - **Solutions:** Simplify user interfaces, provide clear step-by-step guidance, use consistent and intuitive design patterns, and allow users to customize the density of displayed information.

Fictional case study: customizable AR overlays in a museum

Consider a prominent art museum that introduced an AR smartphone application to enrich the visitor experience for everyone, particularly those with hearing impairments. The primary goal was to provide in-depth context for each exhibit, allowing patrons to scan paintings or sculptures for supplementary information, historical background, and curator insights in real time. By engaging with these digital overlays, guests could explore hidden narratives behind each piece and cultivate a deeper appreciation for the museum's collection.

However, some users may struggle with visual information. The following strategies highlight best practices for designing AR interfaces that provide users with equivalent alternatives:

- **Customizable text and visual elements**:
    - **What it is**: Providing options to increase text size, change colors or contrast, and toggle on/off certain overlays.
    - **Why it matters:** Users with varying levels of vision or color perception can tailor the interface to their preferences, ensuring key information remains visible and legible.
- **Multimodal feedback:**
    - **What it is:** Augmenting visual cues with audio prompts, haptic vibrations, or sign-language video embeds for context.
    - **Why it matters:** Deaf or hard-of-hearing users benefit from textual or sign-language equivalents, while users with visual impairments may rely on voice-guided instructions or distinct haptic alerts.
- **Layer toggling and simplified views:**
    - **What it is:** Allowing users to enable or disable specific "layers" of content (e.g., location markers, captions, sign-language overlays) based on personal needs.
    - **Why it matters:** Reduces on-screen clutter and cognitive load, giving all visitors the ability to focus on the details most relevant or accessible to them.
- **Consistent, clear labeling:**
    - **What it is:** Ensuring that each interactive element – buttons, AR triggers, pop-ups – uses descriptive labels, text alternatives, or voice annotations.
    - **Why it matters:** People using screen readers or sign-language features can quickly understand the context of each overlay, promoting smoother navigation and comprehension.
- **Adaptable interaction methods:**
    - **What it is:** Supporting alternative inputs – such as voice commands, touch gestures, or eye-tracking – rather than relying solely on pinching, tapping, or swiping.
    - **Why it matters:** Users with mobility impairments or those who cannot perform certain gestures can still interact meaningfully with AR elements.

By integrating these customization strategies in AR applications, creators can widen and enhance their engagement and deliver a more inclusive experience for visitors.

Adaptation and customization in AR is just one way to create more accessible digital experiences. Now, let's take a look at another powerful tool in this effort, AI, which is revolutionizing how accessibility is implemented at scale.

Leveraging AI to automate the inclusion of accessibility practices

Artificial Intelligence (AI) is everywhere. We see it in personalized ads on social media, facial recognition for unlocking our phones, voice assistants that read our schedules, and image recognition that helps us tag photos. It's also reshaping our day-to-day tasks, from sorting through countless emails and organizing messy files to generating creative suggestions and even automating routine paperwork.

AI is revolutionizing accessibility by identifying common – but not all – barriers with greater flexibility than traditional automation. When combined with agentic frameworks, AI can detect frequent issues such as unlabeled buttons, incorrect heading structures, and low-contrast text while generating standardized reports with actionable remediation steps. Real-time feedback and automated suggestions during development enable engineers to integrate accessibility earlier in the software lifecycle, reducing costly fixes later. While AI cannot replace human oversight, its continuous monitoring and data-driven insights make accessibility efforts more proactive, scalable, and effective.

Integration of AI to support accessible practices presents a unique set of challenges for creators of applications and for users of the content contained therein. For example, AI-powered applications intended to assist visually impaired users must be both forward-thinking and ensure compatibility with existing guidance for compatibility with assistive technology (for example, screen readers). Similarly, it is best practice to implement interfaces, even those with significant incorporation of AI, as simply as possible and avoid overwhelming complexity.

Decisions made during the design and development of AI-driven accessibility features – from the choice of user interface elements to the methods of data processing – can either facilitate or hinder inclusivity. It is crucial to recognize and understand the varied abilities and needs of potential users from the outset.

Building inclusive AI: predictive text, algorithmic bias, and equitable design

AI technologies, while transformative, risk marginalizing disabled users if design and development exclude persons with disabilities. Let's take a look at three key areas – predictive text, algorithmic bias in voice recognition, and equitable AI designing – including their challenges and possible solutions.

Predictive text

Predictive text algorithms often "correct" nonstandard spellings from users with dyslexia, leading to errors that can distort critical messages. For example, consider a dyslexic professional whose email is misinterpreted by autocorrect, resulting in unintended substitutions that impair communication. One potential solution is to expand training datasets to include greater diversity in language patterns and to enable customizable autocorrect settings.

Algorithmic bias in voice recognition

Voice assistants often misinterpret the non-standard speech patterns of users with speech impairments, which can cause practical issues with the intended operation of such tools. If a user with a speech disorder experiences repeated failures in recognition by the smartphone's assistant, it could ultimately lead to real-life scheduling problems. Addressing this challenge involves incorporating a wider range of speech samples into training data and refining recognition algorithms, along with providing updates that improve accuracy for atypical speech patterns.

Equitable AI design

AI systems that rely heavily on visual cues can exclude users with visual impairments, a shortcoming observed in early navigation apps. For instance, when a navigation app offers no non-visual interface, it becomes nearly unusable for visually impaired users. The solution lies in integrating universal design principles, such as auditory feedback, tactile responses, and customizable interfaces.

Fictional case study: AI-powered predictive text for users with motor disabilities

Imagine a pioneering software company introduced an AI-powered predictive text system designed to be inclusive of individuals with motor disabilities. This system utilizes machine learning algorithms to anticipate and suggest the next words or phrases based on the user's typing patterns and contextual usage.

By analyzing user inputs and examining the context of conversations, the inclusion of AI is able to significantly reduce the number of keystrokes required and enable users to communicate more efficiently and with less physical strain.

Additionally, the AI-powered predictive text system can adapt to the unique typing habits of each user, providing personalized suggestions in line with the user's established patterns. This implementation has the potential to facilitate smoother communication for users.

The following accommodations highlight key strategies developers can use to enhance accessibility, inclusivity, and user-friendliness in AI implementations:

- **Simplified interaction models:**
  - **What it is:** Designing AI interfaces with straightforward and intuitive interaction pathways that minimize complexity.
  - **Why it matters:** Simplified interaction models make AI systems easier to navigate for users with cognitive disabilities or those who may become easily overwhelmed by overly complex interfaces, ensuring a more accessible and user-friendly experience.
- **Multilingual and multimodal support:**
  - **What it is:** Providing support for multiple languages and interaction modes, such as voice, text, and gestures.
  - **Why it matters:** Catering to diverse linguistic backgrounds and preferred interaction methods makes AI technologies more accessible to a broader audience, including non-native speakers and users with different communication preferences.
- **Contextual assistance:**
  - **What it is:** Implementing AI-driven contextual help and guidance that adapts to the user's current task and environment.
  - **Why it matters:** Contextual assistance provides timely and relevant support, making it easier for users with disabilities to navigate and utilize AI systems effectively without feeling overwhelmed or lost, thereby enhancing the overall user experience.

- **Error prevention and recovery:**
    - **What it is:** Implementing AI systems that anticipate potential user errors and provide easy recovery options.
    - **Why it matters:** Proactively preventing errors and offering simple recovery mechanisms helps users, especially those with disabilities, to navigate AI systems more confidently and reduces frustration, thereby improving overall usability.
- **Continuous user feedback and iteration:**
    - **What it is:** Implementing mechanisms for users to provide ongoing feedback and using this input to refine AI features.
    - **Why it matters:** Regular feedback from users with disabilities ensures that AI accessibility features remain effective and relevant, allowing for continuous improvement based on real-world usage and evolving requirements.

When select accommodations are embedded into AI-driven systems, individuals with disabilities are likely to engage with technology without encountering additional obstacles caused by the absence of adaptive strategies.

Next, let's explore how to mitigate algorithmic bias in AI-driven interfaces.

Mitigating algorithmic bias in AI-driven interfaces

Algorithmic bias arises when AI systems produce skewed or unfair outcomes due to prejudiced assumptions embedded within the machine learning processes. These biases often mirror societal inequities present in the training data, leading to discriminatory results that disproportionately affect communities of color as well as other user groups.

An AI-powered speech recognition tool, trained predominantly on data limited to a specific demographic, for example, may underperform for users with speech impairments or those from linguistic backgrounds not reflected in the dataset.

Mitigating bias requires a comprehensive strategy that includes diverse data acquisition, transparent algorithm development, ongoing monitoring, and inclusive testing practices in order to enhance the equity of technology and also foster trust and inclusivity within communities.

Data bias and representation

AI models trained on biased data can inadvertently perpetuate existing inequalities, leading to unequal performance across different user groups. In 2018, it was revealed that facial recognition technology used by several law enforcement agencies, including the San Diego Metropolitan Transit System, exhibited significant biases against individuals with darker skin tones (`https://techxplore.com/news/2023-09-skin-color-bias-facial-recognition.html`). The result was inequitable performance of the technology across all user groups.

If AI systems inherit these biases present when training datasets, this can cause them to perform poorly for certain demographics. This lack of representation can lead to exclusionary outcomes where the needs of specific user groups are not adequately met. More broadly, this can also result in users receiving inaccurate information, and undermines users' ability to interact with digital content effectively and independently.

Here are some strategies to address data bias:

- **Regular bias audits:** Conduct evaluations using fairness metrics, anomaly detection, and demographic analysis to identify disparities in model performance across user groups.
- **Diversify training datasets:** Incorporate data that reflects the diversity of users – this may include, but is not limited to, various disabilities, cultural backgrounds, and linguistic variations.
- **Inclusive development teams:** Engage individuals from disability communities in the development and testing phases to ensure solutions address real-world challenges faced by users.

Risk of ableism and overdependence on tool implementation

Ableism and unconscious biases within development teams can influence the design and functionality of AI tools, leading to solutions that prioritize efficiency over genuine inclusivity. This can manifest in assumptions about what users with disabilities need, potentially overlooking critical aspects of accessibility that require more nuanced approaches.

Businesses increasingly rely on automated accessibility tools, overlays, and AI-driven solutions to achieve compliance, but overdependence on these technologies introduces significant risks. Automated checkers and overlays often provide a false sense of accessibility, missing critical usability issues that impact people with disabilities and impact the business with additional costs in development, remediation, and implementation of its services.

As an example, accessibility overlays, while well-intentioned, are argued by critics to enable ableist practices by providing superficial fixes that do not address underlying accessibility issues and sometimes interfere with existing assistive technologies. Overlays could obscure important content, disrupt keyboard navigation, and fail to provide meaningful improvements for screen reader users.

Implementations like this can result in frustration, decreased trust, and a sense of exclusion among users who feel their unique challenges are not being acknowledged or addressed.

Here are some strategies to address prejudicial thoughts and ableism:

- **Cultural competency training:** Provide training for development teams on cultural competency and the diverse experiences of individuals with disabilities. This fosters a more empathetic and informed approach to tool design.
- **User-centric engagement:** Adopt a user-centric design philosophy that involves continuous engagement with users from various disability communities.
- **Inclusive policy development:** Develop and enforce policies that explicitly address and counteract ableism within the organization. This includes setting standards for inclusive language, design practices, and equitable treatment of all users.
- **Regular diversity assessments:** Conduct assessments to evaluate the diversity and inclusivity of development teams and their practices. Incorporation of a diverse workforce is fundamental to mitigate the risk of ableist assumptions and promote a broader range of perspectives.

Best practices for mitigating algorithmic bias

The following strategies outline essential methods developers can implement to assess and reduce potential biases in AI-driven assistive technologies.

- **Diverse and representative data collection:**
    - **What it is:** Gathering training data that accurately reflects the diversity of the target user population, including variations in race, gender, age, disability types, and linguistic backgrounds.
    - **Why it matters:** Ensuring that the dataset encompasses a wide range of user demographics helps prevent the AI model from developing skewed or biased patterns and establishes more equitable performance across different user groups.

- **Bias detection and auditing:**
    - **What it is:** Implementing systematic procedures to identify and measure biases within AI models, using tools and frameworks designed for bias detection.
    - **Why it matters:** Regularly auditing AI systems for bias enables developers to pinpoint and address unfair patterns before impacting users and maintains the integrity and fairness of the technology.
- **Transparent algorithm design:**
    - **What it is:** Developing AI algorithms with clear, understandable structures and providing documentation that explains how decisions are made.
    - **Why it matters:** Transparency in algorithm design allows for greater accountability and facilitates the identification of potential biases, enabling stakeholders to trust and verify the fairness of AI-driven interfaces.
- **Inclusive testing and validation:**
    - **What it is:** Conducting comprehensive testing with diverse user groups, including individuals with various disabilities, to ensure the AI system performs equitably.
    - **Why it matters:** Inclusive testing helps uncover hidden biases and usability issues, allowing developers to refine the AI models to better serve all segments of the user population.
- **Continuous monitoring and feedback loops:**
    - **What it is:** Establishing ongoing monitoring processes and feedback mechanisms to track AI performance and user experiences over time.
    - **Why it matters:** Continuous monitoring ensures that any emerging biases or performance disparities are promptly addressed, maintaining the long-term fairness and effectiveness of AI-driven assistive technologies.

When these strategies are implemented in AI-driven interfaces, individuals with disabilities can engage with technology in a fair and unbiased manner.

Fictional case study using real-world dataset: FairFace – reducing racial bias in facial recognition for assistive applications

A technology company developing an AI-powered facial recognition system for accessibility purposes identified significant racial biases in its initial model, which frequently misidentified individuals from minority backgrounds.

To combat bias, the company integrated an inclusive dataset known as FairFace, which represented a more balanced representation of various racial groups. Through extensive testing with diverse user groups and incorporating feedback from affected communities, the company successfully enhanced the system's accuracy and fairness. This initiative not only improved the reliability of the facial recognition technology for all users but also demonstrated the company's commitment to equitable AI practices.

To further mitigate bias, the company implemented a multi-step approach that included reweighting training samples to ensure proportional representation, employing adversarial debiasing techniques to reduce discriminatory patterns in model predictions, and leveraging **explainable AI (XAI)** methodologies to analyze decision-making transparency. It also implemented system checks provided through human feedback, in which user corrections and annotations were integrated into model training. This iterative refinement process ensured that more biases could be identified and addressed in real-world scenarios, and direct input from affected communities could be included to shape the system's evolution toward greater fairness.

As a result, the system achieved greater accuracy across all demographics, significantly reducing misidentification rates. User trust increased with improved transparency and responsiveness, while the human feedback mechanism ensured ongoing refinement, making the model more adaptive and equitable over time.

By integrating fairness, transparency, and ongoing enhancements, companies can refine AI systems for greater accuracy and equity, establishing a responsible framework for technological advancements that lead to both innovation and inclusion.

Summary

In this chapter, we explored the accessibility challenges and opportunities within emerging technologies – VR, AR, and AI – unpacking how thoughtful design choices can make these experiences more inclusive. From adapting VR environments for sensory and mobility needs to customizing AR overlays for clarity and usability, we examined real-world strategies that enhance digital equity. We also tackled AI's role in automating accessibility, mitigating bias, and ensuring fairness in assistive technologies.

In the next chapter, we'll shift our focus to exploring how companies and developers have successfully implemented these accessibility principles to build truly inclusive products.

Now, if you'll excuse me, I have a cello-playing bird to conduct.

10

Foundations of Accessible Design System Patterns

By Charlie Triplett

Foundations for structures must provide a robust platform for everything being built on top. Design system documentation often relies on such metaphorical concepts, and when we apply physical ideals to accessibility, we find a deeper meaning that will affect our techniques and priorities.

Any physical structure must be designed and built with an understanding of all forces that will be placed upon it – not just the aesthetics. If a structure is built in an earthquake zone, naturally, it must prioritize geotechnical stresses – but it still has to resist other natural events such as rain, snow, ice, and wind. A foundation that can resist an earthquake but erodes in a rainstorm is not a success.

Likewise, a shallow design system that considers only one assistive technology without first digging deeper to support all the surrounding and intersectional disabilities will be just as unsuccessful – this chapter will help you avoid that trap.

Screen readers are the most intensive and dazzling piece of assistive technology often demonstrated by accessibility experts, and for good reason. Many are still unaware that people who are blind can even use computers or smartphones, and this revelation has a great impact.

However, the power of screen readers has led to an over-reliance on them as the primary reason for accessible practices, leading to an over-emphasis on supporting screen readers themselves as meaning "accessibility," instead of considering people with disabilities.

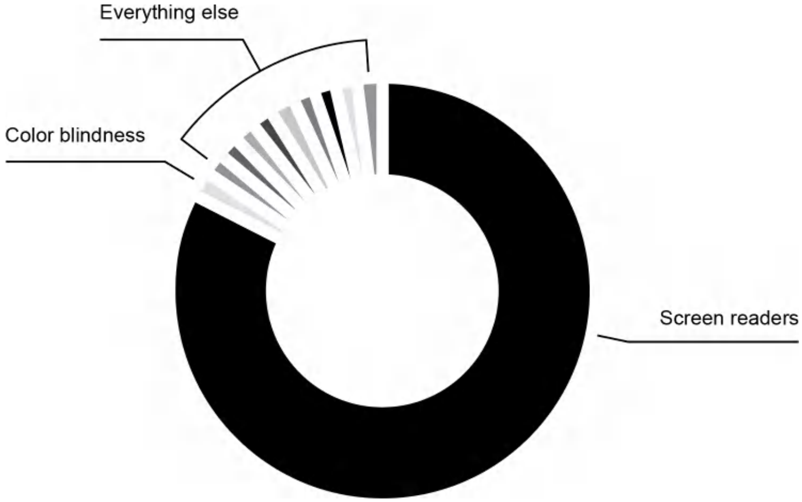


Figure 10.1: How teams who've only been introduced to screen readers think they need to expend their efforts

But, of course, there are people with disabilities other than blindness who need us to remove barriers, and they're not even using any assistive technology such as screen readers or relying on devices such as keyboards:

- People who would say they don't even have a disability encounter barriers daily when text is too small to read or text contrasts too little against its background, especially when increasing the default font size has no effect
- People having color vision deficiency encounter obstacles when colors alone, such as red and green, indicate meaning
- Full-screen video takeovers can cause motion sickness in people with vestibular disorders
- Someone who is deaf or hard of hearing can prefer a transcript of a video, rather than consuming an entire video and watching synced captions
- Someone who doesn't speak another language fluently can have difficulty completing complex forms
- People with speech disabilities might seek a live chat feature, rather than phoning customer service

To actually create an accessible design system, it must remove barriers for all people, laying a broader and thoughtful UX foundation for those using screen readers and those with intersectional disabilities.

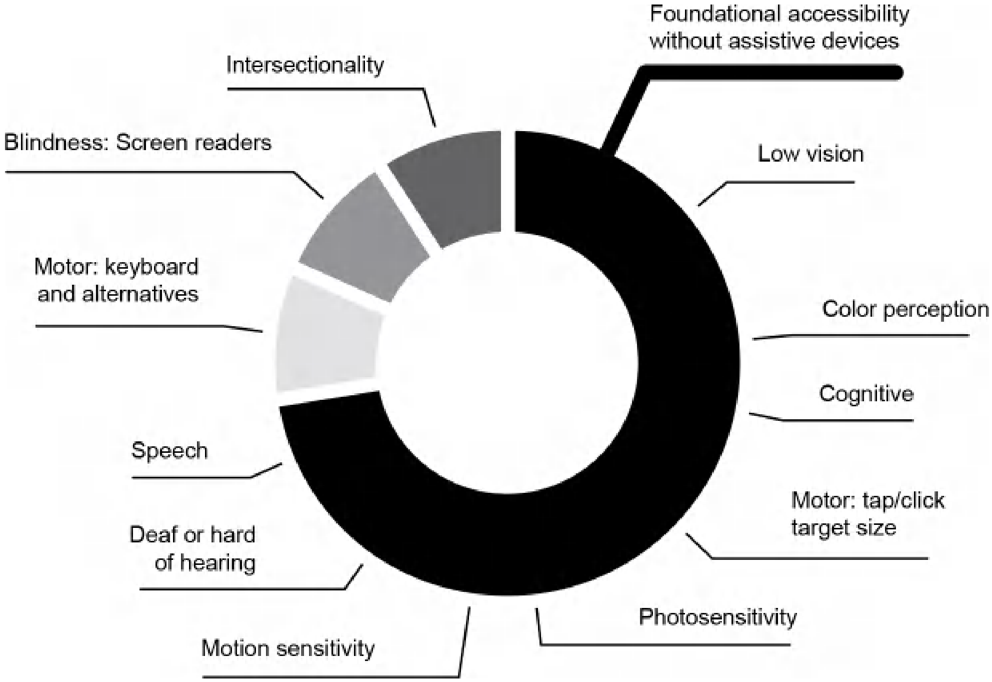


Figure 10.2: A more effective approach to design systems, thinking about people first, not just technology

This chapter breaks down accessibility challenges in everyday design patterns and components from a people-first perspective. It covers practical guidance for structuring design systems in a manner that nudges consuming teams to design and build with accessibility by default, without having to patch in accessibility later.

First, we will consider how to design for people using no assistive devices or simply device accessibility settings:

- Low vision
- Color perception
- Deaf or hard of hearing
- Cognitive disabilities
- Motor: designing for imprecise movement
- Motion sensitivity

- Photosensitivity
- Speech disabilities

Then, with these considerations complete, we will look at designing for people using assistive technology:

- Motor: designing for assistive devices and software
- Very low vision or blindness
- Intersectionality of disabilities

By the end of this chapter, you will recognize potential barriers people with disabilities face, possess a framework for evaluating what people need and how to plan a robust design system – all in the right sequence to maximize inclusion.

Low vision

When all people are taken into account, beginning with low vision, it becomes clear we are actually designing to remove accessibility barriers for over half the population.

According to The Vision Council, 63.7% of adult Americans wear prescription eyeglasses, and in 2018, 92.4% of US Medicare beneficiaries aged 65 years or older reported that they used glasses. These same people wouldn't say they have a disability or need accessibility help because their eyeglasses or contacts (forms of assistive technology) correct their vision.

As we age, our eyesight changes too. Around half of people use dynamic text to bump up the text size on their mobile devices. These are people who might not be using reading glasses and certainly wouldn't think of themselves as having a disability.

The Dutch Appt Foundation survey (`https://appt.org/en/stats`) found the following:

- 45% of the Dutch iOS users surveyed have one or more accessibility settings (such as enlarged text size) activated on their phones
- 15% use two or more accessibility features on iOS
- 61% of the Dutch Android users surveyed have one or more accessibility settings activated on their phone and 25% use two or more accessibility features

By accommodating low vision first, valuable affordances cascade to the rest of the design system for people with other disabilities. The design system must be robust enough for dynamically resized typography, define the intent of color and typography assets, be flexible enough to reflow based on browser settings, and recognize alternative ways people experience the UI.

Planning for dynamic typography

People must be able to change the default size of text up to 200% on the web, or as the settings allow on the specific platform, browser, or operating system. Notice that this is not the same as using the *Cmd+* or *Ctrl+* keys to zoom in with the browser.

If fonts cannot increase or the layout doesn't accommodate larger text, not only does the interface become unreadable or even appear broken but anyone translating the UI into another language will also encounter a broken layout (for example, translating from English to French).

Text must be able to increase in size without breaking its container or becoming partially or fully hidden, independent of an arbitrary specific layout height or width being available. All objects in the system must accommodate a robust variety of typographic conditions (including changing the default font family itself). When creating design patterns, demonstrate how the feature will adapt to larger text.

Enabling web browser zoom

People with low vision often rely on browser zoom (using keyboard shortcuts like (*Cmd+* or *Ctrl+*) to enlarge everything on a page.

The best way to accommodate this is to document and guide teams consuming the design system to use responsive mobile-first design and development practices. Rather than thinking of zoom as a separate feature, it helps to understand that zooming essentially resizes the entire viewport as if viewing the content on a smaller screen.

Specifically, the *WCAG SC 1.4.10: Reflow* guideline generally expects a UI to function without scrolling in two different directions in a viewport of 320 CSS px for a vertically scrolling page, or a height of 256 CSS px for horizontal scrolling content. For example, this means when your responsive mobile-first design begins at 320 px as the smallest viewport, your UI conforms without any additional effort.

Delivering intentional typographic patterns, not just assets

While a design system should give as much creative license to consuming teams as possible, that same license will create inaccessible patterns if left unchecked. There is no more important guidance than the usage of typography, and because of its nuance and technicality, design systems must be prescriptive about their intended usage.

Font scaling and font size: understanding the difference

When planning typographic patterns, it's crucial to understand how font scaling and font size differ – and why this matters for accessibility:

- **Font size** refers to the numerical value assigned to text, such as 16 px or 1 rem. It's a code-level instruction that tells the browser or device how large the text should be.
- **Font scaling**, on the other hand, is about how the typeface itself renders visually at that size. Factors such as *x*-height, stroke weight, and letter proportions affect how large or small a font actually appears on the screen. Two different typefaces assigned the same font size might appear significantly different in optical size because of these characteristics.

Design systems must account for both font size and font scaling to ensure text remains readable and meets accessibility standards. A brand font that renders smaller than a standard font at the same size may need to be assigned a larger size to compensate for its visual scale.

For example, commonly available Arial renders at a scale true to its assigned pixel line height. 16 px Arial will optically measure 16 px of height from the top of the tallest letter to the bottom of the lowest descending letter. (This may vary slightly based on the browser's rendering engine.) However, a different typeface might render optically smaller despite being assigned the same font size. In the following example, even though the code stipulates a font size of 16 px, the custom brand font with poorly adjusted scaling renders noticeably smaller.

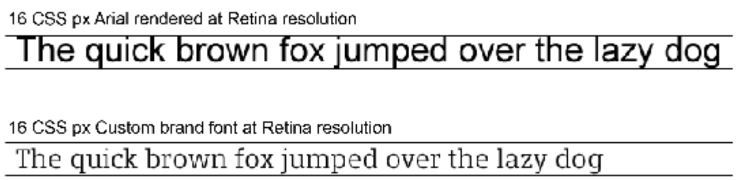


Figure 10.3: Mismatched scaling affects font size rendering

Text contrast

All people using a handheld device will experience situations where visibility is poor. For example, when a smartphone is used outside in bright sunshine, the screen is comparatively dim and difficult to read. Likewise, when a screen is dimmed to conserve battery, low-contrast text and graphics become obstacles for everyone.

This table lays out the generally accepted color contrast requirements for text. As detailed previously, because different typefaces can render at different scales, the CSS pixels must be interpreted optically, rather than simply through their defined pixel size.

Optical CSS pixels	Weight	Ratio
<24 px	Normal	4.5:1
>=18 px	Bold	3:1
>=24 px	Normal	3:1

Figure 10.4: General text size and contrast guidance

For people with low vision, sufficient contrast of text against its background is crucial to removing barriers to consuming content. Text that meets contrast requirements in one context may present a barrier in other contexts, as in this example:

This 18px dark gray [#222222] text contrasts 16:1 against pure white [#ffffff], easily meeting contrast requirements.

This 18px medium gray [#767676] text contrasts 4.5:1 against pure white [#ffffff], barely meeting contrast requirements.

This 18px medium gray [#767676] text contrasts 3.9:1 against light gray [#eeeee], not meeting contrast requirements for small text.

This 18px bold medium gray [#767676] text contrasts 3.9:1 against light gray [#eeeee], meeting contrast requirements for bold text.

This 24px medium gray [#767676] text contrasts 3.9:1 against light gray [#eeeee], meeting contrast requirements for large text.

Figure 10.5: Contrast and sizing examples

Informative graphics contrast versus stroke width

Informative graphics, such as icons or status indicators for controls, must accommodate low vision as well when they identify the name, role, or state of UI components. Solid thick shapes, such as an alert icon, must only meet a 3:1 contrast ratio, but a minimalist hamburger menu icon using 1 px strokes must meet a 4.5:1 contrast ratio:

Stroke width	Ratio
1 px	4.5:1
>=2 px	3:1

Figure 10.6: General graphic stroke width and contrast ratio

Guide color choices, don't just deliver a palette

Teams consuming the design system for their product design and development should be given text color and background pairings that meet contrast standards. While multiple tools exist to perform mathematical analysis of color pairings and produce a table of acceptable combinations, not every combination is visually desirable. Deliver examples of acceptable patterns and tasteful usage so consuming teams can make good choices.

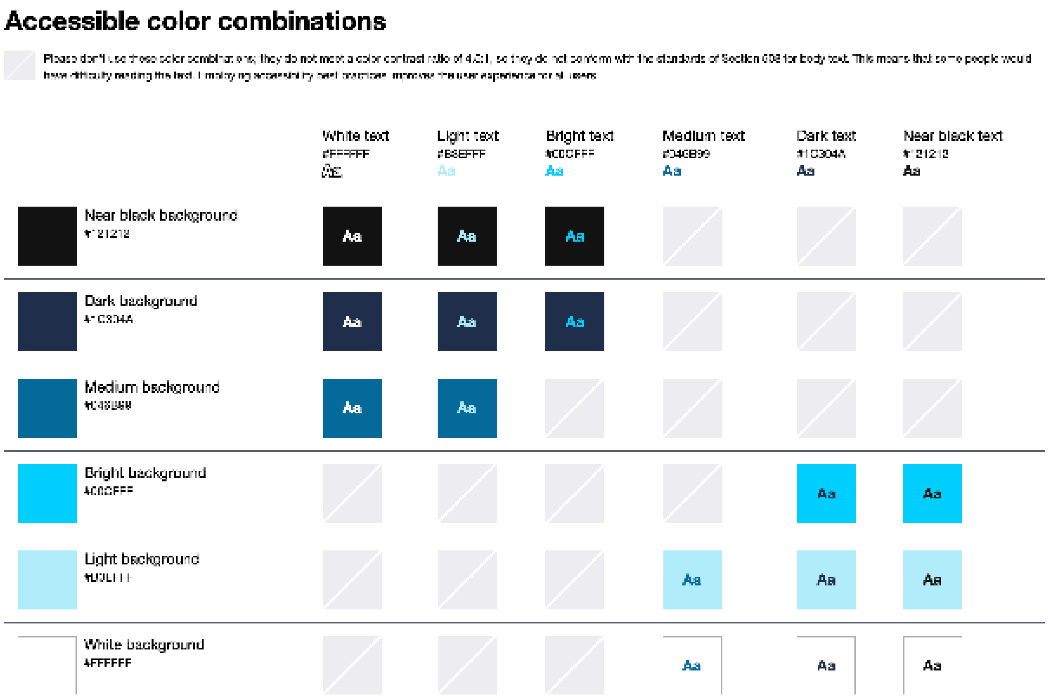


Figure 10.7: Automated tools such as toolness.github.io/accessible-color-matrix can be used to identify accessible color pairings tables.

We must avoid 100% white paired with 100% black. For people with astigmatism, this stark contrast can cause text to appear uncomfortably vibrant, especially with white text on a black background.

Affordances for magnification tools

People with very low vision may also use magnification software, allowing them to magnify very small portions of their screen. On handheld devices, this often means pinching/zooming in.

Seeing only a small portion of the screen at a time means people with low vision will struggle to find content and controls arbitrarily scattered across the page.

While some controls are unquestionably grouped with their labels, such as checkboxes or radio buttons, toggle switches and expanders are often separated to opposite sides of the content, creating barriers for people with low vision.

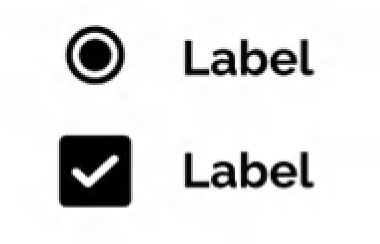


Figure 10.8: Checkboxes and radio buttons are almost universally arranged with a control and label in close proximity to each other

Meanwhile, these toggle switches and expanders have been allowed to fill the available space, no doubt satisfying to the design manager but problematic for people with low vision:

Figure 10.9: These toggle switches and the expander all create barriers for people with low vision

Someone viewing their screen using magnification will struggle to understand the relationship between these labels and controls.

Placing all controls and indicators in direct proximity to the text label helps everyone. If you must place the control indicator opposite the label to the right, set a maximum width for the component.

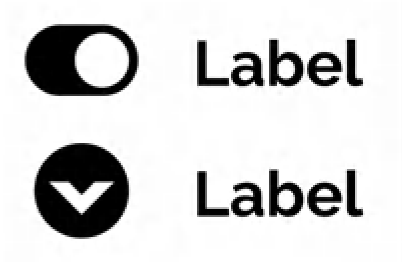


Figure 10.10: Placing the toggle switch and expander control indicators in proximity to their label helps everyone

With these barriers removed for people with low vision who may be using magnification, browser zoom, or just changing default font sizes, a design system will become a foundation that creates a better experience for all.

Color perception

People with color blindness, also known as color vision deficiency, have difficulty or an inability to distinguish between certain colors. This impacts roughly 1 in 12 men and 1 in 200 women (8% of the population) and means that tasks involving color perception, such as identifying traffic lights or sorting clothes, can be troublesome.

When it comes to design, using green and red to indicate the success or failure of a process becomes a barrier for people who are unable to perceive the difference between those colors.

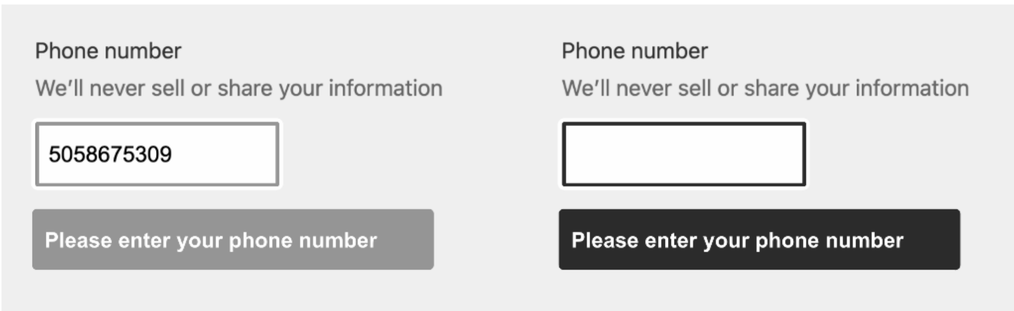


Figure 10.11: Without the aid of color, it's impossible to discern the difference between these two error messages

Instead, include text to convey a status such as success, alert, or warning. Spend at least as much time on content strategy as you do on picking the perfect alert icon. While intuitively meaningful icons can be used, it's rare that they're more useful than actual words.

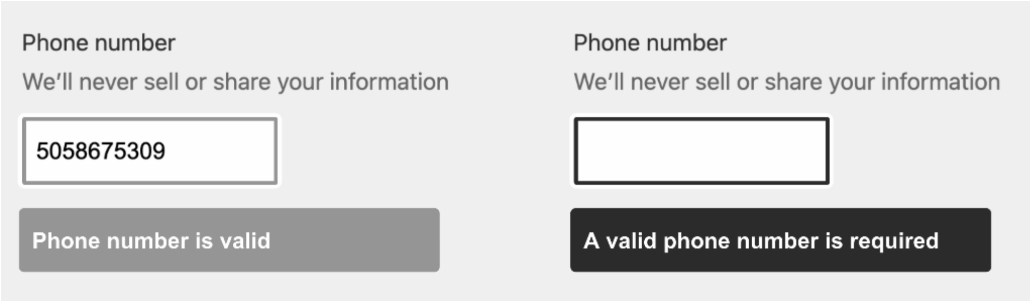


Figure 10.12: These success and error messages do not depend on icons or color alone to convey meaning, and instead use actual words to help discern the differences between the error messages

Once again, by ensuring that the design system doesn't rely on color but rather uses text to convey meaning, it adds affordances that will cascade downward for people with cognitive disabilities and people using screen readers.

Deaf or hard of hearing

Design system audio and video features must provide accessibility through two key features:

- Closed captions
- Transcripts

Both features benefit not just D/deaf and hard-of-hearing users but also people with other lived experiences, such as non-native speakers, neurodivergent users, and people in distracting or quiet environments.

Closed captions

Closed captions are synchronized text representations of spoken words, sounds, and important audio cues in a video. They allow people who cannot hear audio – or who struggle to process it in certain environments – to follow along with the content. Closed captions are essential for D/deaf and hard-of-hearing users but are also widely used by people in noisy spaces (such as airports), people who are non-native speakers, and even by those with sensory sensitivities.

Design systems should do the following:

- Make captions toggleable. Users should be able to easily turn captions on and off within the video player.
- Ensure captions are synchronized. They must appear in real time, accurately representing spoken content and important sounds (e.g., laughter, door slamming, etc.).
- Use legible styles. High-contrast text, adjustable font size, and good placement options improve readability across devices and screen sizes.

Transcripts

Transcripts are full-text versions of audio or video content. Unlike captions, they aren't necessarily synchronized with the media but provide the complete text that can be read independently.

Transcripts benefit a broad range of users:

- Screen reader users can navigate the transcript more quickly than scrubbing through audio
- Non-native speakers can reference the transcript for unfamiliar words or phrases
- People with low bandwidth can read the content without waiting for the video to load

Design systems should do the following:

- Provide a clearly labeled link or button to download or view the transcript
- Place the transcript near the video player or in a dedicated, easy-to-find tab (never hidden behind obscure menus)
- Include meaningful headings and structure in transcripts to aid scanning and comprehension

As you can see, captions and transcripts are more than just technical features – they are bridges to inclusion, connection, and equitable access to content. By embedding these affordances into design systems with thoughtful placement and intuitive toggles, teams create products that meet real needs in real contexts.

Cognitive disabilities

Cognitive disability refers to a range of differences that affect an individual's ability to process, learn, remember, and solve problems.

Use plain language in plain sight

First of all, you should recognize that people aren't always rich in time, attention, and focus to consume, search for, understand, and effectively use your products. Anyone can be affected by a hectic schedule, a noisy environment, or drowsiness from a cold medicine. Likewise, people with cognitive differences may need extra time to complete tasks or recall information.

If people can't find, consume, and understand the content, they can't use your product. Through style guides and design pattern examples, encourage teams to avoid jargon, idioms, marketese, and specialty phrases.

Discourage consuming teams from hiding important information behind tooltips, dialogs, or expanders. Hidden information will not be as readily consumed as useful information placed in visual proximity to decisive UI features.

Affordances for text inputs

Cognitive load increases for all people when all text fields are the same size. Instead, define rules for sizing text fields based on the expected input.

For example, if a field asks for the birth year, the field should be wide enough to contain four characters, but not more. This gives all people the ability to intuit the function of controls and input fields at a glance, both increasing the speed at which everyone can fill in form data while simultaneously reducing errors.

Encourage autocomplete

People invented computers to do repetitive work, and inputs using autocomplete attributes allow just that. When forms don't use or allow autocomplete, the friction to filling in forms correctly increases for all people. For people with motor disabilities who are using switch devices or alternative navigation, this can be an especially tedious and frustrating barrier.

Discourage distractions

People with difficulty focusing need distraction-free interfaces. For them, an autoplay video or background animation that cannot be quickly paused can mean a UI is unusable. Meanwhile, popups trying to upsell in the middle of a transaction can cause irritation (even for those who have no problem focusing), and push users elsewhere. To solve this, it's simple: if it's annoying, it's not accessible. Set strict guidance on how dialogs, modals, and alerts can be used.

By prioritizing plain language, sensible input design, and minimal distractions, design systems can foster a more inclusive experience that empowers people with cognitive disabilities to participate fully.

Motor disabilities: designing for imprecise actions

Design can accommodate people with motor disabilities who simply have difficulty precisely interacting with the UI but aren't using other assistive technology.

It's not difficult to understand that bigger buttons are easier to click on for everyone. People with difficulty using a mouse or tapping their screen with precision will encounter barriers when buttons are too small or packed too closely together.

To help, encourage mobile-first design with generously sized controls. While it's common for consuming teams to feel pressured to make densely featured interfaces, the design system examples and documentation must provide examples supporting an accessible target size.

To meet minimal WCAG AA target size requirements, objects must be 24x24 px in size with a 4 px margin. A more generous size of 44x44 px meets WCAG AAA and HIG requirements, and Android Material Design calls for a 48x48 dp minimum size.

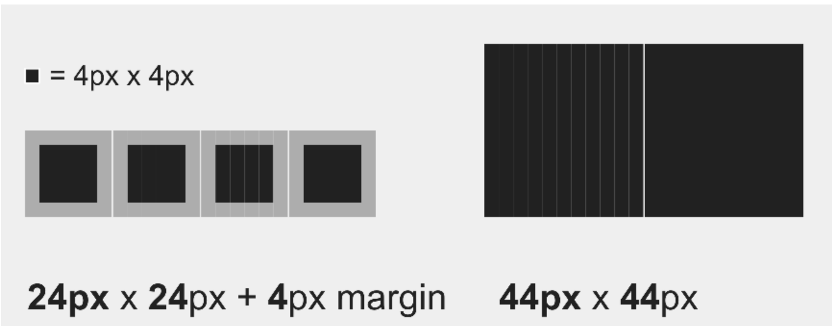


Figure 10.13: WCAG AA minimum target size and AAA target size

For HTML-based controls such as checkboxes, radio buttons, expanders, and switches, the entire label must also be clickable, not just the control itself.

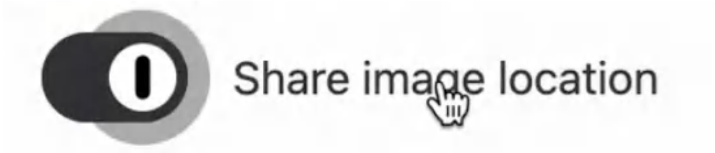


Figure 10.14: Web-based controls such as switches must activate when clicking on the label and the control itself

By designing generous touch targets and ensuring entire labels are clickable, teams can remove barriers for people with motor disabilities – creating interfaces that feel intuitive and usable for everyone. A design system that champions these affordances not only meets accessibility standards but also delivers a more inclusive experience.

Motion sensitivity

People with vestibular disorders can be made ill by sweeping animations or videos. Designs using parallax scrolling or auto-playing full-screen videos can literally make someone sick for hours, causing dizziness, nausea, or disorientation.

Respecting reduced motion settings

Start by documenting how autoplay video and animations must respect the device's **Prefers Reduced Motion** settings. This system setting, available in most operating systems, is detectable in browsers and by native mobile applications, indicating a preference for less motion or animation. For example, when a user has enabled this setting, design systems should disable parallax effects, slow down or remove transitions, and pause automatic animations unless user-triggered.

Plan ahead for changes to motion

The design system must document how video or animation will react to being turned off. For example, if a hero banner features a looping background video, provide a static fallback image with similar branding or messaging. For micro-animations, replace them with subtle fades or highlights that convey the same meaning without inducing motion sickness.

Feature easy-to-find controls

Any autoplay videos must include intuitive, easy-to-find controls for pausing or stopping playback. Avoid burying these controls in nested menus – ideally, position them near the video itself with clear labels such as **Pause animation** or **Stop motion**. For background animations (e.g., moving gradients or background effects), offer toggles or switches so users can easily disable them if needed.

Design systems that respect motion sensitivity build trust with users by prioritizing comfort and well-being. By offering flexible motion settings, fallback visuals, and easy-to-find controls, teams create interfaces that empower people with vestibular disorders to engage safely and confidently.

Photosensitivity

People with photosensitivity carefully manage their exposure to digital environments to reduce the risk of seizures or illness. Triggers include flashing lights (such as strobe effects) or certain high-contrast visual patterns, such as stripes, grids, and checkerboards. For some, even subtle flickers can cause dizziness, headaches, or disorientation.

Set clear guidelines

The design system must establish clear guidelines for all animations and videos, ensuring that any content qualifying as flashing does not exceed three flashes per second. This aligns with *WCAG 2.2 Success Criterion 2.3.1* and helps protect users with photosensitive epilepsy.

Provide warnings and options

Video or animation features must consider where to intuitively place warnings within the interface if photosensitive content is present. Placement will naturally vary depending on the platform – web, smartphone, tablet, or TV – but should always be consistent and easy to find.

Plus, you should also:

- Include a clear, readable label such as **Warning: Flashing content may affect photosensitive users**.
- Offer toggles to opt in or out of high-motion effects. For example, allow users to disable parallax scrolling, background videos, or dynamic gradients.

Test for safety

The design system should document how to test content using automated tools such as the **Photosensitive Epilepsy Analysis Tool** (**PEAT**) software from the Trace Center at the University of Maryland (`https://trace.umd.edu/peat/`). Additionally, consider collaborating with accessibility specialists or user research participants who have photosensitivity to ensure a real-world, lived experience perspective. Work with users with photosensitivity to understand their experiences and needs, and validate solutions.

Additional tips

As well as the previous information, consider these tips:

- Avoid high-contrast flashing between light and dark colors
- Test not only isolated features but also complex interactions, such as transitions that combine multiple visual effects

- Incorporate a consistent approach across all design patterns, including marketing, onboarding, and interactive features

By building these practices into the design system, teams ensure that people with photosensitivity can navigate digital experiences safely and confidently – without the risk of triggers.

Speech disabilities

When we think inclusively, we can understand that many people without disabilities dislike voice calls. It takes time, it can become a cumbersome social interaction, and if the conversation is difficult or unpleasant, such as canceling a service or concerns a delicate personal matter, a phone call can be an uncomfortable and anxious experience to be avoided.

Voice conversations and systems using voice recognition often create barriers for people with speech disabilities. If the only way to place an order or receive customer support is through voice channels, this represents a barrier. Engaging with people with speech disabilities early in the design process helps identify these barriers and co-create alternative solutions.

While WCAG rules don't call out specific success criteria covering speech or voice disabilities, we can derive an approach of providing consistent alternatives:

- If teams are using the design system to add product features such as voice assistants and speech-enabled apps or will be relying on **interactive voice response** (**IVR**) phone systems (e.g., "Say the name of the department you're trying to reach"), the design system must guide teams to provide non-voice alternatives at every step.
- Add clear visual and text-based alternatives – for example, a chat interface, SMS option, or email support – so users can choose their preferred communication method.
- Prioritize asynchronous options so users aren't forced into real-time interactions that can cause stress or anxiety. Offering accessible alternatives also benefits users who might be in loud environments, have poor reception, or simply prefer typing.
- Provide consistent, easy-to-find options to switch between voice and non-voice channels. For example, during an IVR phone call, allow users to press a number to connect with a text-based support channel or request a transcript.
- Include explicit guidance on the role of captions or transcripts within voice-enabled products. For example, voice assistants should be able to display text-based confirmations and results. This empowers people with speech disabilities to understand and navigate the system without relying on voice alone.

At this point, having removed barriers for people with different disabilities such as low vision, cognitive, motor, and photosensitivity, our design system is now primed to offer affordances for people who use assistive technology such as keyboards, switch devices, and screen readers.

Motor disabilities: designing for assistive devices and software

People with severe motor disabilities use assistive technology ranging from keyboards to more sophisticated and often customized switch devices. It's crucial that the design system encourages consuming teams to consider these kinds of assistive technology as it is a prerequisite for screen reader accessibility.

It's important to understand the fundamental difference between navigating a UI using a mouse versus using assistive technology such as a keyboard. People using a mouse are in control, like driving a car around a city: free to drive around the screen in any order. Using a keyboard is more like being on a train, experiencing the content and controls in a linear guided journey. Each stop on the route is mandatory.

Single point activation

We will all be in situations where complex operations on touch devices aren't possible because of the environment (such as walking in a busy street) or because of distractions (such as carrying a fussy animal). But also people using a keyboard or switch device will not be able to perform complex gestures such as swiping or dragging with precision.

In these cases, the ability to perform an operation with a single tap increases everyone's ability to complete a task. Provide a single-point method of operation for compound features such as navigating draggable controls or (if unable to omit them from the system) carousels.

Accommodate landscape, portrait, and upside-down orientations

Everyone experiences the need to consume content on a handheld device in multiple orientations. For example, you may be browsing content on a plane using those seatback pop-out tray thingies. However, people with motor disabilities may use a handheld device that is permanently mounted and unable to be rotated. If the UI isn't usable in all orientations, this becomes a barrier.

So, ensure that all content is consumable in every orientation. Anything covering the content of the screen such as popups, sticky floating ads, or menus must not create barriers to consuming content in any orientation.

Focus style definition

Focus isn't just about people using a keyboard to navigate their devices. Text inputs must have a focused state for all people, even those tapping their screen or using a mouse. This focus state lets all people understand that the field is focused and available to receive input. Unfortunately, many developers are taught to hide focus styles, degrading the experience for all people.

Figure 10.15: Text field focus outlines the input itself, even for people using a mouse

People using a keyboard, switch device, or alternative navigation rely on the interface visually indicating which control is currently focused and ready for interaction. Only interactive controls should be keyboard-focusable. Headings, sections, or landmarks must not be keyboard-focusable.

The design system must guide teams to a uniform focus pattern. If a design system ignores focus styles or simply leaves it to defaults, not only will focus often be absent but it will also be unpredictable.

While it may be desirable to create branded custom styles, relying on browser default colors and actively applying browser default styles to controls can achieve a high degree of consistency, especially when products rely on third-party plugins.

> Native smartphone applications should almost always use default styles, as they're deeply tied to the operating system, and are modifiable by contrast settings for low vision.

Defining focus styles includes where to place focus outlines. For example, following default HTML focus patterns, this group of checkboxes places focus on the styled checkbox itself:

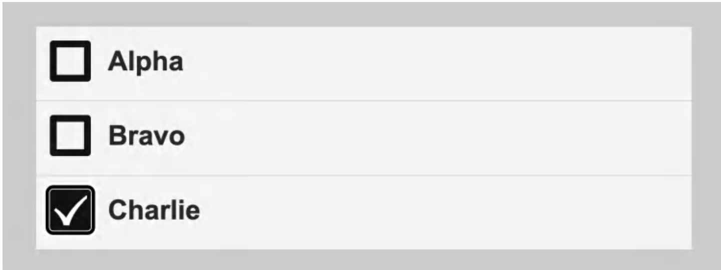


Figure 10.16: These WCAG-compliant checkboxes add a focus outline around the styled checkbox

This checkbox group places the focus around the entire label:

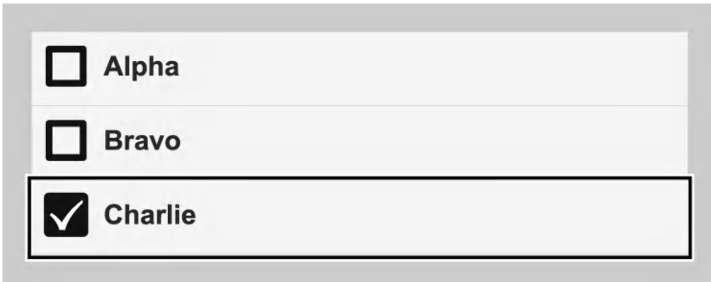


Figure 10.17: These WCAG-compliant checkboxes add a focus outline around the checkbox label

But don't overdo it. Remember, the purpose of focus indication is to let the user know which control is ready for interaction. For example, do not wrap a text input, label, and helper text in a focus outline. The surrounding text is not a control, so it must not appear as focusable.

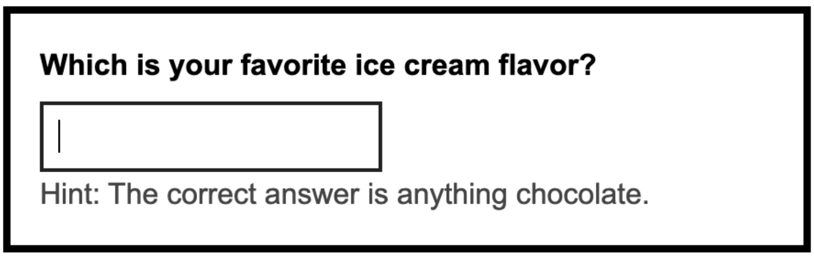


Figure 10.18: Labels and descriptive text must not be outlined on focus

In short, just let the browser do what it already does: indicate a focus on the interactive focusable element, not the surrounding text, card, or section.

Accommodate alternative navigation

Some people rely on entirely different ways of navigating digital interfaces, such as voice control, eye tracking, or switch scanning systems. These alternative methods are used by people with very limited mobility, speech disabilities, or those who simply prefer hands-free interaction.

People using voice control (such as *Dragon NaturallySpeaking* or built-in mobile OS commands) need to be able to refer to what they see on the screen in a predictable, verbal way.

For that reason, UI controls and text must be clearly and uniquely labeled. You can think of words like a picture made of letters that have meaning. For example, instead of using a trash can icon button for canceling an order, you can use the word **Cancel**. Instead of a paper plane icon button to send a message, you can use the word **Send**. Both of these examples allow someone using voice controls to perform this action in a single command.

So, design systems for websites and applications cannot skip accommodating keyboard accessibility, not just because it is clearly a prerequisite for people using screen readers but also because barriers can exist for keyboard users that aren't barriers when using a screen reader. Next, let's look at how design systems must remove barriers for people who are blind or have very low vision.

Very low vision or blindness

The previous sections have removed many of the challenges of designing for people using screen readers, who will benefit from all of the affordances already discussed:

- Flexible layouts accommodating all devices, orientations, and viewport sizes work well for people using a screen reader because they may be using a handheld device or be unaware of their desktop screen's resolution or browser window size. In fact, they may not have a desktop monitor at all – why bother if you can't see it?
- Meaningful heading structure allows the screen reader to skim through content, surfacing useful information.
- Transcripts make it easier to consume audio content, and audio descriptions make it possible to consume videos containing important visual information.
- Content that doesn't rely on color alone to convey meaning will use words to convey meaning.
- A language that doesn't rely on nuanced marketese or jargony language can be pronounced correctly by screen readers.

- Autocomplete helps reduce errors for everyone while speeding up form completion.
- Focusable controls in meaningful sequence lead people using a screen reader on a guided linear journey through the UI.

These considerations lay the groundwork for creating design systems that not only comply with accessibility requirements but also truly empower people who are blind or have very low vision. Now we'll explore how design systems can further support these users by addressing headings, focus management, alternative text, and assistive technology considerations.

Choose a non-semantic typographic naming convention

The design system must adopt a naming convention that doesn't rely on numeric identifiers to indicate size and weight but focuses on the intended usage and style of the text. These simple examples allow for any style to be applied to any heading level. In this manner, `h1` can be the smallest type visually, but maintain its meaning to people using a screen reader.

The following styles can be applied to any heading, regardless of its heading level:

```
.text-normal {
      font-size: 1rem;
}
```

```
<h1 class=text-normal>
      My coffee dashboard
</h1>
```

```
.text-medium {
      font-size: 1.5rem;
}
```

```
<h2 class=text-medium>
      My favorites
</h2>
```

```
.text-large {
      font-size: 2rem;
}
```

```
<h2 class=text-large>
      Featured roasts
</h2>
```

```
.text-huge {
      font-size: 4rem;
}
```

```
<p class=text-huge animate>
      Coffee jitters!
</p>
```

Figure 10.19: Semantically agnostic styles

Provide annotation tools (but don't overdo it)

Some degree of standardized and meaningful accessibility annotation must be available for designers to communicate unclear names, roles, or states for UI components. The way accessibility is communicated to engineering teams must not be an afterthought.

In the absence of explicit annotations, developers will seek markup clues in design styles and token names. If a heading is using a style named `heading-1`, they will often wrap the content in an `<h1>` tag, even if the heading is not intended as the primary title of the page.

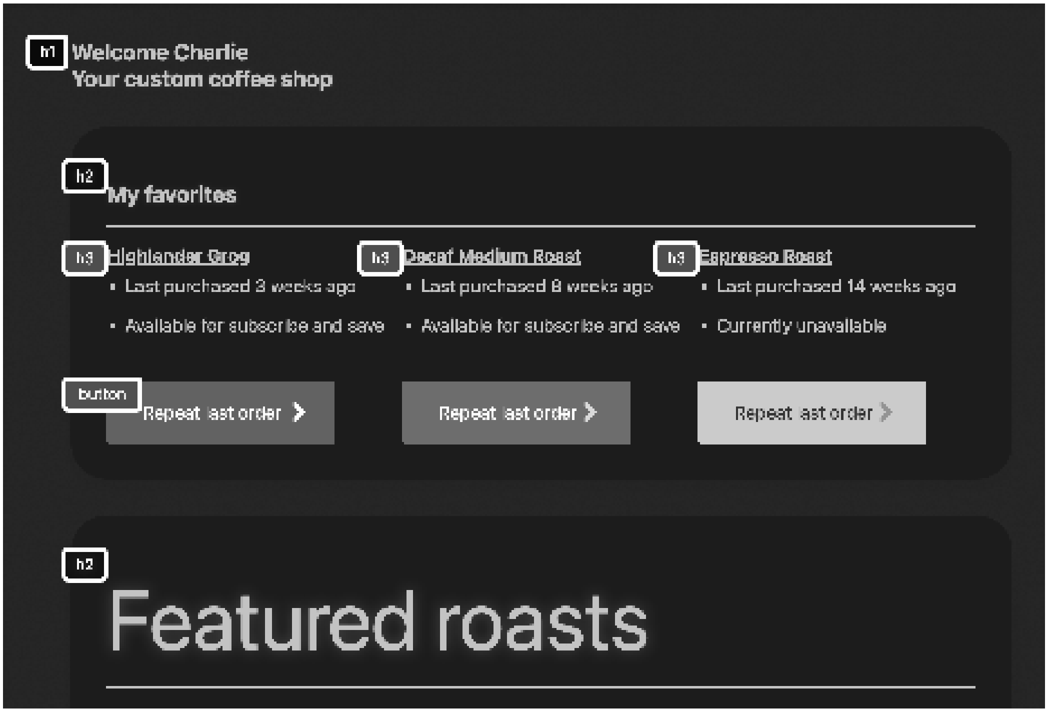


Figure 10.20: Simple annotations for headings and buttons

Whether it's a sticker sheet library, a specific software plugin, or detailed inline instructions, the ability to add meaningful annotations for everyone is a must-have.

Work directly with your developers and establish sustainable expectations for annotations and their usage. Is it helpful for them to see it inline, next to, or on top of the design artifact? Would they prefer a separate annotated view?

Recognize where annotations are and aren't necessary. If UI designers are embedded directly in cross-functional engineering teams and communicate directly with Android, iOS, and web developers on a daily basis, annotation needs may be minimal. On the other extreme, if the UI design team is siloed and simply tosses artifacts downstream into a waterfall production environment, more verbose annotations will be required.

A minimal annotation library should include options for the following:

- Landmarks (header/banner, navigation, main, footer/content info)
- Heading levels (`h1`, `h2`, `h3`, etc.)
- Alternative text names for images
- Control roles (checkbox, radio button, switch, etc.)
- Autocomplete attributes (`name`, `address-line1`, `address-level1`, etc.)
- ARIA labels (for naming sections, forms, or navigation)

While annotations are useful, they mustn't be superfluous. For example, in a design artifact where the small screen mobile layout and the large screen desktop layout are side by side, it may only be helpful to annotate the mobile layout, only noting differences or additional features in the desktop layout.

Alternative text for informative versus decorative images

To achieve a good experience for people using a screen reader, the design system must encourage consuming teams to add alternative text descriptions for informative images and discourage alternative text descriptions for decorative images.

Consuming teams need clear guidance on what constitutes a decorative or informative image and how to describe either. If the design system doesn't set patterns for alternative text usage, consuming teams will either ignore alternative text, leaving out crucial visual content contained in photos and graphics, or overuse alternative text for decorative icons, leading to repetitive or nonsensical experiences.

By default, UI icons delivered by the design system must be name-agnostic and considered decorative with no alternative text. Design systems typically supply iconic UI imagery that practically never needs a description. Even with a typical set of UI icons, it's impossible to write a single alternative text description for an image to achieve good UX in all circumstances.

For example, if this warning icon is assigned an alternative text of **Warning**, that is not appropriate to all contexts. "Warning" implies something destructive is occurring. But, naming the icon something softer, such as **Alert** or **Reminder**, might not be strong enough for a non-reversible decision such as account deletion.

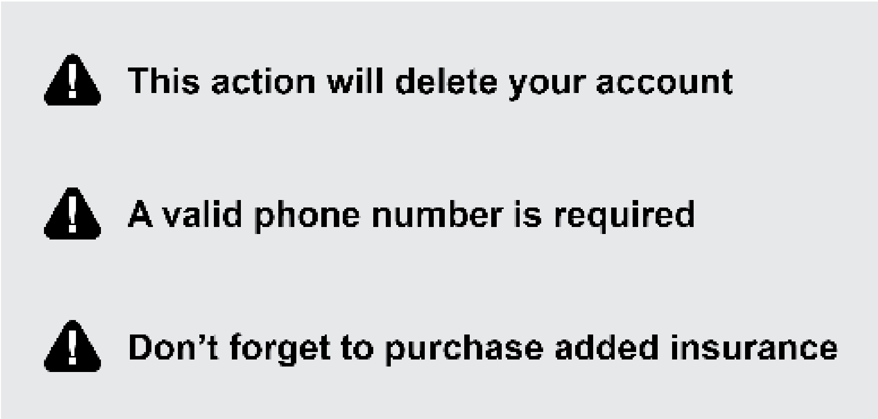


Figure 10.21: Assigning a singular alternative text value to this icon is impossible

Audio description for videos

Video with audio description includes an alternative audio track describing content necessary for understanding video content that isn't spoken. Incorporate an audio description feature for video content that requires an audio track.

Ideally, provide a simple and intuitive way for users to turn audio descriptions on or off – such as a clearly labeled button or menu option within the video player. Don't bury this feature in the settings menus where it's hard to find.

Incorporating audio descriptions effectively not only makes video content accessible to more people but also aligns with best practices for inclusive design.

By integrating meaningful headings, accessible typography, annotations, alternative text, and audio descriptions, design systems can empower people who are blind or have very low vision to fully engage with digital experiences. Thoughtful, inclusive design benefits everyone, making information easier to find, understand, and navigate.

Intersectionality of disabilities

Out of practicality, accessibility in design is discussed in the context of a single disability or assistive technology, but disabilities are much more complex when people with multiple disabilities are considered.

A design system that accommodates dynamic text sizing and consistent focus styling can accommodate someone with low vision and a severe motor disability. People who find animations or motion distracting may use magnification tools to focus, framing out distracting features. People who have cognitive disabilities may use a screen reader or closed captions to help with reading comprehension or understanding audio content. Real-world scenarios are rarely so neatly separated. Involving users with intersecting needs – such as someone who uses a screen reader and also experiences motion sensitivity – helps teams understand and design for complex, overlapping challenges.

When design systems are built with flexibility, clarity, and alternative ways to interact, they can better accommodate this complexity without requiring exceptions or additional customization. It's important to remember that many people experience multiple disabilities at once, such as low vision and motor impairments, or hearing loss combined with cognitive challenges. A design system that anticipates these overlapping needs – such as ensuring high-contrast typography, consistent focus styles, clear labels, and single-point activation – creates a more equitable and consistent experience for everyone.

Features to omit for everyone with disabilities

Accessibility for design systems doesn't just mean making everything accessible; it means deciding what components and patterns to omit. Some common UI components can be left out of a design system to encourage consuming teams to seek better solutions. Collaborating with users to understand why certain components create barriers is essential – this helps design systems evolve to support inclusive experiences rather than simply removing features.

Tooltips

Important content required to understand the UI should be displayed for all people, without being hidden. Indicative of poor UX design, tooltips ask people to perform additional cognitive tasks to understand the UI. Tooltips are an especially common barrier for screen readers because there is no common convention to define their name, role state, or keyboard action.

Instead, encourage consuming teams to write the UI to be intuitive and self-explanatory.

Custom-styled select (aka listbox/combobox/dropdown)

Precisely zero good reasons exist to replace the native HTML `select` element in forms. An HTML native `select` element is accessible and provides all the features necessary to have a pop-out list of options.

A custom dropdown cannot replicate the robust functionality of a native `select` element in every platform (macOS, Windows, iOS, Android, and ChromeOS) because each has slightly different keyboard actions, touchscreen gestures, and semantics.

Further, it is expensive to build correctly and, without consistent maintenance, will inevitably become inaccessible to people with motor disabilities using keyboards or switch devices and to people using screen readers.

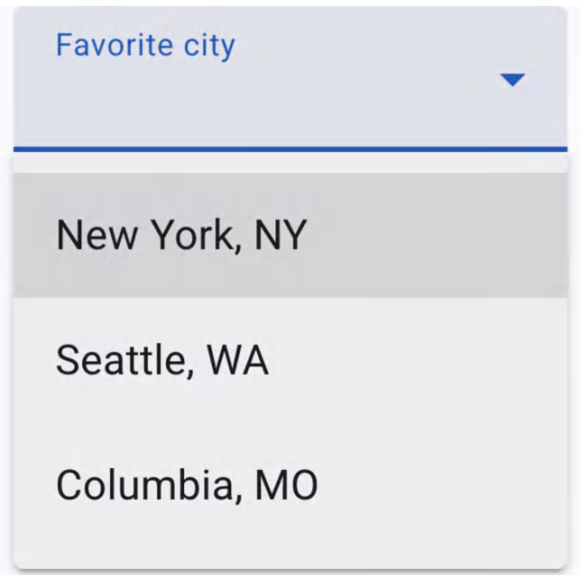


Figure 10.22: Angular Material custom dropdown

The native HTML `select` element can be completely styled in its collapsed state while the options surface styling is provided by the operating system and does not require redesign, though future CSS proposals may allow for more styling options.

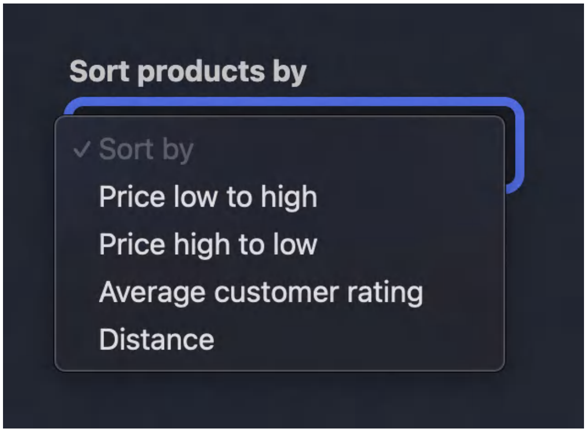


Figure 10.23: The native HTML select surface in macOS

Carousels

Carousel slideshows are often a compromise between stakeholders who couldn't decide what content should lead, so they choose all of it. This results in hostile user experiences that rarely receive engagement beyond the first slide.

Future browser features may include native-like support for accessible carousel patterns, but this doesn't mean it's the best way to convey content. Even accessible carousels deliver an additional cognitive load for all people who must learn how to navigate its special compound UI and consume its unique content. Is it swipeable? Do the arrow keys operate it? How many slides are present? This and more information must be conveyed and understood by the user, and people must be appropriately motivated to engage beyond the initial view.

If carousels are an unavoidable component of the design system, help teams use accessible carousels wisely to eliminate barriers, reduce cognitive load, and add value to the experience:

- Avoid using auto-advance. This creates barriers for people with cognitive disabilities and vestibular disorders. If featured, auto-advance must be off by default and user-controllable.
- Guide teams to favor predictable information. Carousels work better when people can guess what's coming (e.g., shoe colors and product variations). Avoid surprise or disjointed content across slides.

- Set expectations upfront. Clearly present the number of slides, navigation controls, and current interval position for people with low vision and those using a screen reader.
- Carefully consider what is announced by screen readers when the user has interacted with the carousel. For example, it's useful to announce what has changed (slide title and interval position), but not the entire slide if it contains a great deal of information.
- Keyboard and screen reader users must be able to skip the carousel without interacting with every slide.

By anticipating overlapping disabilities – such as low vision combined with motor impairments or hearing loss coupled with cognitive challenges – design systems can foster more inclusive digital experiences. Designing for intersectionality ensures that no one is left behind, creating products that work seamlessly for everyone, regardless of the unique combination of needs they bring.

Summary

This chapter examined how an accessible design system must offer more than bare components. You've learned how a design system that focuses on removing barriers for people with disabilities in the right sequence lays a strong and sustainable foundation for a scalable and accessible design system.

That means beginning with the most common needs such as low vision, color perception, and cognitive disabilities – and building toward more complex accommodations for assistive technology such as switch devices and screen readers.

Further, a design system must be a guide that consuming product teams use to make good decisions consistent with the intent of the system. Integral to the design system must be opinionated guidance about typography, interaction patterns, and what features not to include – such as tooltips and custom selects – so consuming teams aren't left guessing. Treat your system like an intuitive product for consuming teams, and inclusion for people with disabilities like a requirement, not a bonus.

Next up is a guide to accessibility testing – what works, what breaks, and what actually matters across web and mobile. You'll learn how to blend automated checks with manual testing, how to triage what's worth fixing first, and how to fit it all into real-world development workflows.

11

Tools and Techniques for Accessibility Evaluation and Prioritization

By Kai Wong, CPACC, CHES®

Accessibility is about creating digital experiences that work for everyone. Even if a design *looks* accessible, many accessibility issues live under the hood – within the code.

Accessibility testing bridges the gap between design intent and actual user experience. It is an important part of **quality assurance** (**QA**). Just like with any other quality check, evaluating product accessibility involves testing it.

> The **Web Content Accessibility Guidelines** (**WCAG**) is a widely accepted technical standard that can guide digital accessibility testing. You can find the latest version of the guidelines on the **World Wide Web Consortium** (**W3C**) website (`www.w3.org`).

This chapter provides an overview of digital accessibility testing tools and techniques, focusing on web and mobile app accessibility testing, its challenges, and common issues to look out for. You'll learn how to combine automated and manual testing methods, prioritize what to fix, and bring accessibility into development workflows. The goal? To help you and your team test smarter and build more inclusive products.

In this chapter, we're going to cover the following main topics:

- Selecting appropriate automated testing tools
- Performing effective manual accessibility audits
- Interpreting and prioritizing accessibility test results
- Integrating accessibility testing into development workflows

Selecting appropriate automated testing tools

Who doesn't love a good shortcut? I know I do, so let's talk about automated testing tools. Using automated tools is an easy, quick way to do accessibility checks, popping out results faster than toasting bread. In seconds, they can find accessibility issues such as missing alternative text, poor color contrast, missing accessible names for form elements, and absent document language.

There are several automated tools available in the market today. When choosing automated testing tools for accessibility, consider these key factors:

- **User interface**: Is the tool intuitive and easy to navigate? Is it accessible?
- **Reliability**: Are the test results consistent and reproducible?
- **Ease of use**: Does it require training, or is it easy for a beginner?
- **Browser and device support**: Does it work on the browsers and devices you need?
- **Maintenance**: Is the tool updated regularly?
- **Cost**: Is the tool free or does it come with a cost? If you find a tool that requires a subscription or fee, test it with the free trial first. Explore your options. Find what fits your needs and budget. In most cases, the free version will do.
- **Standards**: Does the tool check against the standards at the conformance level you aim for, such as WCAG 2.2 Level AA?

Most automated accessibility tools are available for free directly in developer tools, as mobile applications, or as browser extensions. The following table compares free, commonly used automated tools:

Tool	Platform	Features	Drawbacks
WAVE by WebAim (`www.wave.webaim.org`)	Web	• A browser extension • Opens in a side panel. • Scans a full page in its load state • Shows page structure, such as headings, and uses on-page icons to highlight results	• The page can get cluttered with on-page icons • Does not indicate the impact level of issues found
axe DevTools by Deque (`www.deque.com/axe/devtools`)	Web	• A browser extension • Opens inside the browser's Developer Tools • Scans a full page in its load state • Issues can be highlighted one at a time • Relevant requirements and impact levels are listed for each flagged issue • Premium upgrade options include partial page scans, user flow scans, report exports, and guided manual testing features	• Small learning curve to using the tool at first • Many features require upgrading to the paid version
Lighthouse by Google Chrome	Web	• No download is required since it's built into Chrome's Developer Tools • Easy-to-use user interface • Uses the same library that powers axe DevTools, a trusted framework and widely respected resource for accessibility testing • Provides a score out of 100, giving a quick, high-level assessment of a page's overall accessibility status • Can also check for performance, SEO, and best practices	• A high score may mislead people into believing a web page is fully accessible • Sometimes it is one version behind the latest available version of axe DevTools.
Accessibility Inspector by Apple	iOS and macOS mobile apps	• Developer Tool available in XCode • Provides raw accessibility information about specific elements within a mobile app • Offers element inspection capabilities, VoiceOver simulation, and screen-level audits	• Since it's part of Xcode, this tool is primarily used by developers during the development phase. It's less convenient for non-developers or testers who aren't working within Xcode. • A limited number of tests are covered.
Accessibility Scanner by Google	Android mobile apps	• An app available to download from the Google Play Store • Can scan a single screen or a series of screens	• A limited number of tests are covered.

Figure 11.1: Free automated testing tool options

The tools in *Figure 11.1* were selected based on the estimated number of users and personal experience. There are many other options (especially web tools).

W3C's *Web Accessibility Evaluation Tools List* (`www.w3.org/WAI/test-evaluate/tools/list`) has information on over 65 automated testing tools. You can filter tools by purpose, cost, operating system, language, and standards.

Since many tools are free, there's no harm in using multiple at a time. Double-dip like you're at an ice cream shop! Use as many as you want. Compare results and cross-check for accuracy.

Are they too good to be true? Yes, sort of. Automated tools can only cover about 30–50% of standard accessibility testing needs. Here are a few examples that snuck past automated tools but were later caught by a manual tester or end user:

- Form elements, such as radio buttons, with `null` or `object` in the accessible name but not in the visible button label
- A link that couldn't be activated using the keyboard alone, unless a screen reader was running
- Several important lines of overlapping text on a website's home page when zoomed to 400%

This means that even if a screen passes all automated tool checks, accessibility issues may still exist. That's where manual testing and user feedback come into play.

Many digital accessibility issues are context-dependent and require human review. Automated tools are a solid starting point, but they're just the beginning. Always back them up with manual checks and insights from real users (especially those with disabilities). Thorough manual testing and diverse user feedback are key to creating accessible, user-friendly experiences. Let's take a look at effective manual testing next.

> For a deeper analysis of automated tools and manual testing, check out Adrian Roselli's blog post, *Comparing Manual and Free Automated WCAG Reviews* (`https://adrianroselli.com/2023/01/comparing-manual-and-free-automated-wcag-reviews.html`).

Performing effective manual accessibility audits

Sure, automated testing tools may help you find accessibility bugs quickly but, if you stop there, you are skipping the best part: manual testing. That's right, it's time to roll up those sleeves and get into manual testing.

Manual test methods will uncover issues that automated tools cannot. It's kind of like cleaning up pet fur. One round of vacuuming will not clean everything; you may need to get a lint roller to make sure it is completely fur-free. That's why using multiple tools and a variety of testing methods is critical for ensuring accessibility.

We will use "issues" and "bugs" interchangeably throughout this chapter. Also, the following information applies to both the web and mobile, unless specifically mentioned otherwise.

While I won't go deep into WCAG in this chapter to keep things approachable, most of the testing strategies discussed are related to WCAG standards. For specific criteria or to dive deeper into each testing area, refer to the official WCAG documentation.

Keyboard navigation is key

Some people with vision or mobility disabilities may use alternative input methods, such as a keyboard, to navigate digital content. Websites and mobile apps should be fully usable without touching a screen or clicking a mouse. As a result, the focus order – the order of elements that receive focus when navigating with an alternative input method – and keyboard navigation are essential to test.

If you have a mouse, just let your cat play with it while you complete your manual tests. Meanwhile, you can test the focus order and make sure that all elements can be reached using the keyboard alone.

You can test keyboard navigation on mobile devices using a wireless keyboard, such as a Bluetooth keyboard.

Starting at the top of the web page, use the *Tab* key to move from interactive element to interactive element. Examples of interactive elements include buttons, links, drop-down menus, sliders, toggles, checkboxes, and text fields. Tabbing should move you through the page and show a visible focus indicator (i.e., outline) on the focused element.

Make sure the focus order is logical (which typically means that moving forward with your keyboard matches the order and meaning of the page) and fairly predictable. It should not jump around on the page like burning garlic on a hot stove. In most cases, the focus should move from top to bottom and left to right. If you're tabbing and suddenly your focus does a disappearing act… congratulations, you found a bug!

You can refer to the following table to test keyboard navigation on common web elements:

Element Type	Keys
Button	*Enter* or *Spacebar* to activate
Links	*Enter* to activate
Selection dropdown or combo box	*Enter* or *Spacebar* or *Alt* + *Down* arrow to expand it; *Up* or *Down* arrows to move between options; *Esc* or *Alt* + *Up* arrow to close the drop-down menu
Checkbox	*Spacebar* to select or deselect
Radio button	Arrows to navigate between options; *Tab* to leave the group of radio buttons and move to the next interactive element
Toggle	*Spacebar* (though sometimes *Enter*) to switch it on/off
Sliders	Arrows to increase or decrease the value

Figure 11.2: Common keyboard testing interactions for websites

Mobile devices have their own flavor of keyboard commands. The usual suspects are *Tab* (to move forward), the arrow keys (to move in the direction of the arrow), and the spacebar to activate most interactive elements. If you're working with a specific device, check out the keyboard accessibility settings screen, where keyboard commands are typically listed.

Get friendly with your screen reader

Testing with common screen readers, such as **NonVisual Desktop Access** (**NVDA**) (for Windows, available for free), **Job Access With Speech** (**JAWS**) (Windows, paid), VoiceOver (macOS and iOS, free), and TalkBack (Free, Android), is an essential part of accessibility manual testing. So, fire one up and start exploring!

While testing with a screen reader, be nosy. Explore that page like it's a new town you've been itching to visit for years.

> Desktop screen readers typically operate in two modes: browse mode and focus mode. Browse mode is used for reading static content (such as paragraphs and headings) and focus mode is for interacting with controls, such as form input fields and buttons. Refer to the screen reader's user guide for further instructions.

As you navigate your way through, pay attention to how the screen reader describes each element in focus. Screen reader announcements should be helpful, not hectic or absent; they shouldn't be repetitive or point out any unnecessary junk.

If this is your first trip to screen reader city, you may notice it speaks fast. Don't worry, you can adjust the speed in your screen reader's settings. There's no shame in slowing things down at first. Most daily users use their screen reader at full speed, but if you don't often use one, slow it down to a more comfortable speed.

For interactive elements, such as buttons or links, the screen reader should announce the label followed by the type of element (called its "role"). For example, a button with a text label of "Add to cart" should be announced as "Add to cart, button." A big no-no would be if the visual label says "Add to cart" but the screen reader announces it as "Buy it now, checkbox," which can confuse or mislead users.

Additionally, if the accessibility name does not match an element's text label, it may cause issues for users who navigate by voice. When navigating by voice, the person selects elements by saying the visible labels. If the visible label differs from the accessibility name, the assistive technology may not recognize the element or respond to the voice command. Technology advancements may improve this soon though, so follow the latest operating system and software updates.

Also, look out for other common screen reader announcement issues, such as the following:

- Missing role or element type (i.e., not saying that a button is a button).
- Decorative images are not ignored.
- Improper use of headings (i.e., illogical order or not descriptive).
- In-line error messages are not announced.
- Form fields lack required or optional indications.
- Status messages are not announced. For example, if a form is submitted, there should be a success message indicating it was submitted or saved.
- Focus on management issues that impact the user experience.

Mobile device-specific screen reader tips

Turn on your screen reader in your device's accessibility settings (or ask the default voice assistant), and then use gestures to test a mobile application. Gestures are a common way to navigate a mobile device with a screen reader.

Most devices provide a quick tutorial of common gestures when you turn on the screen reader for the first time. Once the screen reader is running, you can use these gestures.

Common gestures include the following:

- **Swipe right**: Move the focus forward
- **Swipe left**: Move the focus backward
- **Single tap**: Select an item
- **Double-tap**: Activate an interactive element in focus
- **Swipe with three fingers**: Scroll the page in the direction of your swipe (iOS only)
- **Swipe with two fingers**: Same as above but for Android

Using the screen reader and gestures, check that the focus order, screen reader announcements, and core functionality behave as expected.

Check descriptions of images

Automated tools are great at flagging that there's a missing image description – often referred to as **alternative text** (**alt text**) – but they can't judge whether the text is helpful or accurate. That's where manual testing comes in.

Assistive technology should describe meaningful images in a way that communicates the purpose. One way to test alt text is by navigating to the image and observing the screen reader or braille output. Listen to (or read) what's announced and ask yourself: Does this make sense? Does it add value without being overcomplicated?

Pay attention to the length of the image description – it should be detailed but concise (no one needs a novel). If the image contains important text, make sure that text is included in the alt description. If the image is just decorative, the screen reader shouldn't even mention it.

Want to double-check the alt text on a website? Pop open your browser's developer tools, inspect the element, and look for the `alt=""` attribute in the HTML code. The alt text is what is between the quotation marks. If the image is purely decorative, the code should still have `alt=""` but no text between the quotation marks. This tells assistive technologies, "Hey friend, you can ignore me. I'm just here for the visual vibe – nothing more, nothing less!"

Test the page in various states

Don't stop there. There is still more to explore! Are error messages, loading indicators, and changes in content (such as expanding an accordion to reveal text) announced? If not, the experience leaves much to be desired and could leave many end users feeling lost or frustrated.

It's important to manually check the behavior of dynamic content because most automated tools only test the page in the state in which it loads. So, for example, it may not test menus or accordions when they are expanded, when in-line error messages are present, or if a date picker appears after selecting an appointment time.

While navigating with a screen reader and keyboard, explore these additional states. When you encounter a menu, confirm that you can reach all options in the menu without using a mouse and that the screen reader accurately announces all the options.

Zoom in on what matters: your product

People with low vision or other disabilities may use features that enlarge digital text or content. That is why it's important to test your experience at various zoom levels and text sizes. Pinch to zoom on mobile devices or increase the web browser's zoom level to ensure all content reorganizes without overlap, cutoffs, or decreased functionality.

Then, test with large text. Adjust your text size in your device or browser settings and then verify that all text resizes proportionately, no content overlaps, and no functionality is lost.

Color counts

Color vision deficiency is quite common. So, make sure color alone is not used to give instructions or convey meaning. For example, errors in a field should not only be marked in red; they should also have an inline error message announced by a screen reader. That doesn't mean you can't use color, you just shouldn't use only color to indicate meaning.

Also, make sure your digital experiences use sufficient color contrast. Most times, automated tools can catch color contrast issues; however, some may slip by. If you can do a quick visual scan of the page, look for any low-contrast text and non-text elements. If anything on your page looks like it went through the wash too many times (i.e., faded), it's time to chat with your designers.

More essential checks

There are several other things to check manually:

- Easy and correct heading structure
- No quick flashing content
- Consistent navigation
- Animations and videos have a way to pause and play

- Videos have captions
- Audio comes with a transcript
- Session timeout warnings, letting users extend their session

Let's finish by looking at some mobile-specific considerations:

- **Orientation support**: Devices can be used in either portrait or landscape mode. This relates to accessibility because some people may have their phone or tablet in a fixed position, such as mounted to a wheelchair. Test your digital product in various orientations to ensure support. When you turn your phone horizontally, the view of your app or mobile website should seamlessly switch to landscape view. Similarly, when you change your mobile device back to the vertical position, the content should switch back to portrait view. In both positions, the content should not overlap, cut off, or lose functionality.
- **Built-in accessibility features**: There are many accessibility features available on mobile devices, so it's important to test how they work in your app or mobile website. These features are typically available under **Device Settings** or **Accessibility Settings**.

  Some features have slightly different names on iOS versus Android. Refer to the following table for several examples.

Feature	Android name	iOS name
Voice navigation	Voice Access	Voice Control
Text-to-speech	Select to Speak	Spoken Content
Screen reader	TalkBack	VoiceOver
Increase text size	Font Size	Larger Text
High contrast	High Contrast Text	Increase Contrast
Minimize animations	Remove Animations	Reduce Motion
Alternative input	Switch Access	Switch Control
Enlarge screen content	Magnification	Zoom

Figure 11.3: Names of iOS and Android features

As I've said, accessibility is a core part of QA and impacts the user experience. Use various tools and testing methods to evaluate your product's accessibility. Manual testing may be time-consuming (especially when you are still learning), but, like any skill, you'll get faster with more experience. Manual testing is necessary to ensure your product is accessible to people with disabilities. It's all about creating high-quality digital experiences that everyone can use and enjoy.

Accessibility testing will likely uncover a range of accessibility issues that need attention. The next move? Document them in your internal bug-tracking system. Then, analyze, prioritize, and help make things better.

Interpreting and prioritizing accessibility test results

Depending on the complexity and scale of your technology and how far along you are in your accessibility journey, you may find yourself with a handful or hundreds of issues following an accessibility audit.

If you find yourself in a big ol' bucket of accessibility bugs, it's time to stop, drop, and roll up those sleeves (again). You must assess and prioritize the list to develop a clear remediation plan for whoever will fix the issues.

Prioritizing bugs

When prioritizing accessibility issues, consider the severity and potential user impact.

Blockers and serious issues

Focus first on blockers – these are the showstoppers that would prevent an end user from being able to complete a critical task within your digital experience. For example, if your login page crashes when someone uses a screen reader, this is a blocker and should be urgently moved to the top of your list. The same thought process would apply to a "Submit Order" button that is not keyboard-accessible or if an onboarding modal traps keyboard focus without a way to exit.

Once blockers are identified and prioritized, shift your focus to serious issues. Serious issues significantly disrupt or burden the user experience. These are the ones that make your product very difficult to use, but they don't necessarily block an essential function.

Examples of serious issues include the following:

- An image containing the company's vision statement is missing alternative text
- Pre-recorded videos are lacking captions; if the video content is critical for completing a task, it could be considered a blocker
- A text-heavy web page has no headings

Moderate issues

After tackling blockers and serious issues, you can move on to moderate issues. Moderate issues may be inconvenient or noticeable to the end user but do not significantly impact the user experience.

Examples of moderate issues include the following:

- One or two decorative images are announced by a screen reader
- A few instances of text have moderately low color contrast
- A simple, short page lacks headings
- "Required" is announced twice by a screen reader on a required form field

Other factors to consider

Other factors can influence how urgently an issue needs to be addressed. Consider the broader context in which issues exist:

- **Frequency**: How many pages does this issue impact? How many times does it happen on one page? For example, if 1 decorative image is announced on a page, the impact is less severe than if 10 decorative images are announced on a page.
- **User feedback**: Have customers or clients reported a poor experience due to accessibility issues? What is the severity of the issue/issues? Low or high?
- **Location**: Is the issue on one of your most visited pages? Does it affect a business-critical user flow? Or is it happening on an outdated event page with minimal traffic?
- **Scope**: Does the issue appear within reusable components, or is it confined? Global issues – those that impact the same component across multiple pages – should be prioritized early in remediation efforts. A common example would be an accessibility issue within a shared navigation menu. If there's no focus indicator in the navigation menu on the home page, the same issue probably occurs on all other pages where this menu is used. Addressing global issues helps create a more consistent and accessible experience across your digital product.

> Teams without accessibility experience often opt to fix the easy, low-hanging fruit first. This could be due to a few factors: low confidence in tackling the more complex issues, not fully understanding the user impact of certain bugs, or a lack of direction due to a gap in prioritization efforts.
>
> While this approach may help reduce the bug count quickly, the impact on the end user is often minimal. However, low-effort fixes can sometimes drive meaningful change. While small effort fixes can sometimes drive meaningful change, it's important to weigh the potential impact alongside the level of effort. Your prioritization efforts will help with this.

By prioritizing more serious problems first, you can make a substantial difference for the end user by removing barriers and improving the user experience. It can also help you avoid having frustrated customers who cannot use your product due to major accessibility flaws or worse (such as dealing with demand letters and legal troubles).

Dashboards

Remember that big ol' bucket of accessibility bugs? Consider creating a dashboard to help make sense of it all. A dashboard can help you interpret your test results, track remediation progress, and inform your strategy. Common variables to track in your dashboard might include the following:

- Issue type/category (i.e., screen reader issues, color contrast, or keyboard navigation)
- Priority level (often a function of severity of impact)
- Impacted user groups
- Remediation status (i.e., not started, in progress, or resolved)
- User flow
- WCAG success criteria
- Assistive technology

Planning remediation efforts requires time and effort, but it offers tremendous guidance to your company, team, or business. A prioritized backlog and a well-organized dashboard can help move your team from "*Where do we start?*" to "*Hey, we've got this.*" This will also help you set project milestones and deliver the most impactful changes first.

Prioritizing thoughtfully and remediating strategically isn't just good for business; it's also a win for people with disabilities.

With these insights guiding your testing and remediation efforts, you are on the right path. Now, let's shift gears from being reactive to proactive. It's time to get ahead of the game and weave accessibility testing right into your development workflows.

Integrating accessibility testing into development workflows

You know what's better than finding and fixing accessibility bugs? Preventing them in the first place! To do this, you must bake accessibility into your **Software Development Life Cycle** (**SDLC**) processes and make it a part of the culture. Sustainable integration of accessibility into development workflows requires a thoughtful strategy, tools, and some trial and error to find what works for your teams.

Start by defining the standard that everyone will strive for, such as the WCAG 2.2 AA. Get buy-in from leadership and document your decision. Consider creating a policy that does the following:

- Specifies accessibility requirements
- Assigns responsibilities
- Outlines methods for monitoring and enforcement
- Identifies relevant tools and technologies

Policies and standards are a solid foundation, but they will not change behaviors themselves. It's like having an instruction manual for assembling furniture. Sure, instructions are helpful, but without the right tools, commitment, and effort, the end result may not be great. You'll need to integrate accessibility into every part of your development workflows and team culture:

- **Training**: Frequent, targeted training builds awareness and gives people the necessary skills to execute accessibility tasks. Highly engaging and effective training programs can also drive cultural change.

> You can collaborate with your learning and development team to weave accessibility tidbits into existing training programs.

- **Tooling and testing**: As discussed earlier in this chapter, developers and QA teams should use both automated and manual testing methods alongside other code functionality checks. Developers can also use accessibility linters to catch issues before the QA stage.

> To make things actionable, be specific. Instead of simply saying "WCAG 2.2 AA is required," state which tools to use, what to test, and common issues to look out for. It's not uncommon for a QA engineer to say that they "did accessibility" but only ran an automated tool. Often, they didn't realize what was expected. Clear instructions help bridge that gap and ensure thorough testing.

- **Requirements**: Indicate accessibility as a requirement, such as by adding it to the global **Definition of Done** (**DoD**) and/or as part of **Acceptance Criteria** (**AC**). The DoD is a shared agreement within an Agile team that defines the criteria a product increment must meet to be considered complete. AC outlines the standards a user story must meet to be considered complete and functioning as expected from the end user's perspective. Partner with your release train engineers, product and engineering leaders, and scrum masters to make this happen.
- **Documentation**: Document accessibility tool choices, specific testing expectations, and your company or team's technical standard (i.e., WCAG 2.2 AA) in test plan and bug report templates. Create easy-to-follow checklists to help developers and QA engineers test accessibility. Role-based accessibility checklists that specify tool names, design considerations, or critical test steps are helpful. Store guides and checklists in places where developers and QA frequently visit, such as the internal bug-tracking system, intranet, or knowledge base.

A well-designed checklist helps people learn independently and build more inclusive products from the start. But before you start building one, take a moment to think through the purpose and audience.

Here's what to keep in mind when creating a checklist:

1. **Define the audience and purpose**: Who is this checklist for, and what do they need to accomplish?
2. **Make it actionable**: Each item should be a specific task (i.e., *Ensure all form fields have associated labels* instead of *Check form accessibility*).

3. **Give it structure**: Group related tasks together so the checklist is easy to use.
4. **Include helpful resources**: Link to relevant tools, browser extensions, or testing instructions to help people complete each requirement.
5. **Iterate and update**: Improve the checklist over time based on feedback. Make changes as best practices evolve, and use versioning to indicate when updates are made.

> Most people are not accessibility experts. Write checklists and resources using plain language, and demonstrate testing methods in multiple formats (e.g., screenshots, videos, in writing). Help people feel confident, not confused.

However, keep in mind that checklists are just a helpful starting point. As mentioned in *Chapter 2*, accessibility itself is not just a checklist. It's a commitment to quality, usability, and people. After all, digital accessibility impacts both the people who build the technology and those who use it.

If you are in the process of integrating accessibility into development workflows, remember that change is not easy. Everything in this chapter has a tangible impact on both the organization and the customer. We discussed the importance of prioritizing feedback from customers, but don't miss out on the voice of your fellow designers, developers, and QA. Their feedback on tooling, checklists, and processes can lead to tailored improvements. Plus, listening to people builds relationships, trust, and better work environments.

Strive to make expectations actionable and manageable. You can do this by providing developers and QA engineers with the right tools and information to perform accessibility tasks as part of their day-to-day work.

Embedding accessibility into existing development workflows not only enables the creation of more inclusive, user-friendly products but also helps shape a culture of accessibility-first thinking. Ultimately, the end goal is to prevent new accessibility barriers from reaching end users. Find what works for your team, collect feedback along the way, and make iterative improvements.

Summary

This chapter laid out the basics of ensuring your website or mobile app will be usable for people with disabilities, focusing on testing tools and techniques. We explored the dynamic duo of automated and manual testing methods, the importance of prioritization, and how to bring accessibility into development workflows.

The topics covered here are foundational to building a successful digital accessibility program, especially in the area of quality assurance. Effective testing and remediation ensure that inclusive design translates into a high-quality user experience. Accessibility is a journey that requires continuous monitoring, effort, and collaboration across all teams. Whether it is one line of code, one test case, or one bug – every small action adds up.

In the next and final chapter, you will look at how you can embed accessibility strategies into organizations.

12

Building an Inclusive Design Culture: Strategies for Organizations

By Denis Boudreau, CPWA

Here we are, down to the last chapter. Let me begin by repeating a lesson discussed throughout this book: inclusive design thrives when accessibility is embedded in an organization's culture, not when it's treated as an afterthought or a last-minute fix. Organizations that build accessibility into their workflows, track meaningful progress, and invest in continuous learning are better equipped to meet the needs of all users, including those living with various disabilities. Teams that recognize accessibility as a shared responsibility rather than a constraining obligation end up creating stronger, more inclusive digital experiences for everyone. Period.

As a way to wrap up this book, this chapter will explore practical strategies to make accessibility a lasting part of how organizations work. This transformation begins with building empathy through immersive experiences and by helping cross-functional teams understand the real impact of accessibility barriers on the audiences they are expected to serve.

In this final chapter, we will focus on four main strategies to build an inclusive design culture:

- Cultivating empathy through immersive experiences
- Integrating accessibility into design and development workflows
- Measuring and reporting on accessibility progress
- Weaving continuous learning into inclusive design

Whether you're leading an organization, managing a team, or are directly involved in the design and development phases of digital products or services, the strategies presented in this chapter will help you move beyond awareness and straight into action. Teams that embed accessibility into their daily decisions are the ones who successfully create experiences that work for everyone.

After all, as the late John Slatin used to say, "*Good design is accessible design*."

Cultivating empathy through immersive experiences

I want to begin by sharing a few thoughts on empathy, as empathy is one of the strongest drivers for inclusive design. Not the feel-good "let's all be nice to each other" version, but the raw, eye-level understanding that comes from witnessing real people struggle.

When design teams witness end users running into accessibility barriers, disability inclusion quickly goes from a constraining theoretical set of requirements to an eye-opening, meaningful shared responsibility.

Most experts would agree that organizations that actively embrace inclusive design are also the ones who ultimately reach more customers, reduce their legal risks, and stay ahead of changing user needs and expectations. That has certainly been my observation over the years.

And while accessibility training sessions, governance policies, and role-based checklists certainly have the potential to go a long way, truly transformational change doesn't happen through an accessibility audit. It happens through engaging directly with those who have relevant lived experiences and consume the world differently.

Understanding accessibility barriers firsthand

Most professionals in design, development, and leadership roles will probably never use a screen reader. Very few are even likely to one day navigate a website without their mouse or fill out a form using only their voice.

Without such firsthand exposure to alternative, yet valid means of navigating the digital space, it's no surprise that for most people, accessibility feels disjointed from design.

Creative teams that interact with digital products the way people with disabilities do can more clearly see the gaps and challenges introduced by a blatant lack of digital inclusion. This realization leads to more thoughtful and inclusive design decisions, which, in turn, lead to more inclusive experiences online.

A few years ago, I guided a UX team at a major software company that needed to improve the accessibility of their platform but wasn't sure where to start. As part of our engagement, I guided them through a workshop where they spent hours navigating their platform using only a keyboard and a screen reader, capturing observations along the way.

As an experiment, we also provided them with blindfolds, so they could go about uncovering major usability issues that had never been on their radar up until that point. Confusing button labels, missing alt text, keyboard navigation traps, and dozens of other accessibility issues suddenly became painfully apparent.

More than minor inconveniences, these issues were real barriers that prevented real people from using the platform independently. While unnoticed by the UX team up until that point, these issues had nonetheless been plaguing the user experience of people with disabilities for months, if not years.

This was no doubt a jaw-dropping experience for the team, but I knew better, so I had one more surprise in store for them. Once they were done experimenting with alternative ways of navigating their websites, I introduced them to Tim, a blind colleague of mine.

Using the exact same tools (minus the blindfold, which I knew was a misleading idea), my colleague went on to uncover a series of additional problems the team had failed to identify, such as poor use of ARIA landmarks, lack of feedback on dynamic interactions, overuse of ambiguous link text, unclear call to action states, and missing text alternatives to otherwise purely visual cues.

What stood out most to everyone was the nature of the issues Tim identified compared to those they had found. While the blindfolded team uncovered many surface-level barriers, Tim's insights went far deeper. He noticed systemic navigation issues, gaps in feedback, and missing structural elements that only someone with lived experience would know to look for. His muscle memory, mental mapping, and intuitive expectations, shaped by years of using assistive technology, gave him access to problems that simulations simply couldn't replicate.

I knew this was a lesson the team would only understand by going through the motions. Simulation exercises sometimes raise awareness, yes, but they don't lead to sustainable, real-world solutions. They might start the conversation, but they can't replace the insights that come from lived experience because true accessibility comes from listening to (and learning directly from) people with disabilities, proving without a doubt that you could only go so far when "faking your way into a disability."

Why disability simulations fall short

Driven by the eagerness to make a deliberate case for disability inclusion, it's not unusual for well-intentioned brands to try to build awareness and empathy through what we call "disability simulations." As we just observed, this is rarely a good idea, because while disability simulation exercises sound great in theory, they fall short when it comes to what they implicitly suggest about the experiences of those who live with these disabilities.

Some teams will have their employees wear blindfolds to experience blindness, as I did. Other teams will use gloves to simulate limited mobility. My personal pet peeve is when businesses have their people move around in wheelchairs, as if this could make someone understand what it's like to live with limited mobility. At first sight, these well-intentioned initiatives sound like fantastic ideas. But I had learned years before that they weren't, and I knew that the only way this team would understand was to compare their blindfolded experience with Tim's.

In fact, all these exercises typically achieve is reinforcing misconceptions about "how horrible it must be to be disabled," rather than building a real understanding of what it might be like to use the web with a disability. Not exactly helpful. Counter-productive, even.

I also once worked with a marketing team that organized labs where employees wore noise-canceling headphones and blurred goggles to simulate being deaf or blind. When asked what they had learned, the response was predictable: "That was so frustrating and depressing. I can only imagine how hard it must be to be deaf or to be blind!"

And that's exactly what disability simulations get all wrong: because these exercises usually focus on what people can't do instead of focusing on how they interact with the world around them, they leave participants with feelings of pity rather than a deeper understanding of real solutions.

All that simulations can do is scratch the surface of a disabled person's reality, and most blind or deaf people would tell you that this is not how they experience the world, digital or otherwise. Deaf people don't simply struggle to hear; they use captions, transcripts, or sign language to communicate. Blind individuals don't just experience impaired vision; they navigate with screen reader technology, tactile markers, and spatial awareness. They have tools and develop coping mechanisms. They have techniques, often honed by years of lived experience.

Instead of disability simulations, organizations should focus on engaging directly with people who live with those disabilities and have relevant lived experiences. By testing their products and services with assistive technology users and integrating accessibility into every stage of design and development, brands can develop a true understanding and perspective of what the barriers are and how to fix them.

But still, yes, it does begin with empathy.

Building empathy through real engagement

The best way for a team to cultivate empathy is to listen to and learn from the people who have relevant lived experiences with these disabilities. I remember a coaching session with a financial tech company a few years ago, where the UX team was struggling with nailing accessibility in their mobile applications.

Despite meeting most W3C **Web Content Accessibility Guidelines** (**WCAG**) requirements, they were still getting usability complaints. As such, some of their most critical banking features, such as transferring money and consulting PDF statements, were completely unusable with screen reader technology. Their app technically passed most compliance checks, but real users were still getting stuck at key moments due to unforeseen challenges.

Acknowledging that they lacked the firsthand perspective, instead of guessing what might still be wrong, they chose to run usability testing sessions with blind and low-vision customers. A wise decision.

What they learned through that process was far more valuable than what any checklist could have ever provided. It's one thing to follow accessibility guidelines around section headings, but it's an entirely different thing to observe the different ways in which blind people using screen readers might navigate from one section heading to another, or what the cognitive load might actually feel like for neurodivergent individuals interacting with our user interfaces.

After incorporating the feedback gathered from these usability sessions, the team not only fixed those issues but also committed to involving users with disabilities in all future product testing. That shift in their thinking is what ultimately led to a better experience for customers living with disabilities.

The changes also improved everyone else's experience in the process, demonstrating that accessibility best practices not only increase the experience of disabled users but also those who benefit from inclusive experiences that are truly centered around the end users.

Embedding empathy and driving company-wide commitment

Empathy is one of the strongest drivers of inclusive design, but it shouldn't be limited to a one-time workshop or confined to design and development teams. Empathy must be part of how people work, collaborate, and address challenges if we are to truly transform the organization's culture.

One of the most effective ways to start building that empathy is through activities such as "accessibility immersion weeks," where employees and executives spend time navigating their own digital tools using only a keyboard or basic assistive technologies. These experiences are not meant to replicate disability; they are meant to expose barriers in a way that can't be ignored. When senior leaders experience these obstacles firsthand, accessibility stops being a compliance issue and becomes a leadership priority.

But teams don't need to wait for a formal event or initiative to start! Even small, everyday moments, such as trying to complete a task without a mouse or reviewing a design with accessibility in mind, can spark insight and shift mindsets. The key is to build momentum through regular exposure and conversation, not just one-off activities.

That moment of recognition is often where change begins. Leaders who understand the real-life consequences of inaccessible design are far more likely to support meaningful actions and empower others to do the same. That's when accessibility champions begin to emerge, and inclusion becomes part of how teams approach their work, day in and day out.

Organizations that keep accessibility top of mind by building it into design critiques also encourage feedback from users with disabilities. They recognize that inclusive successes create space for continuous progress, stop treating accessibility as extra work, and start seeing it as a marker of quality and innovation.

Inclusion is no longer something they do just to avoid lawsuits; it becomes something they want to do because they understand the value it brings to everyone.

Turning empathy into action

Empathy can go a long way, but some would argue that it's only valuable if and when it leads to real change, not feelings of pity. Organizations that most successfully build accessibility and inclusion into their culture don't stop at raising awareness; they take concrete and measurable steps to ensure the long-term impacts that make a difference in the lives of their customer base.

Over the years, I've led countless successful accessibility training programs that left employees feeling inspired to be part of the solution for disability inclusion. I've no doubt turned hundreds of designers and developers into accessibility believers. But for most participants, once their daily work routine settled back in and the magic of the training sessions wore off, so would their motivation. Why? Because if accessibility never quite becomes a priority and the practice is not built into workflows, then other considerations take precedence.

Experience has taught me that in order to ensure that accessibility remains a permanent part of their process, organizations must introduce policies where every new product release has to pass usability testing with assistive technology users before launch. Forget meeting WCAG. Involving users with disabilities is the one decision that creates real accountability and truly leads to more accessible products across the board.

I currently support another organization that has built an accessibility ambassador program to maintain accessibility efforts across teams and lines of business. Employees from different departments volunteer to be internal advocates, guiding discussions, raising concerns, keeping accessibility top of mind… and learning along the way. Over time, this approach is slowly but surely leading to organization-wide changes. Not so much because their leadership mandated it (they did), but because employees who see the value are choosing to take ownership of accessibility themselves.

These cultural shifts are essential, but they only stick when supported by structure. Once empathy opens the door, it's the systems, processes, and day-to-day decisions that keep it open. That's where inclusive design and development workflows come in, and where accessibility has the best chance to move from intention to implementation.

Integrating accessibility into workflows

Another important concept to keep in mind is that accessibility works best when it's built into every stage of design and development, not rushed in as an afterthought or a last-minute fix.

When teams treat accessibility like policies, usability, or security – something essential, not optional – they reduce costly rework, make smarter decisions, and create better experiences for everyone.

Organizations that succeed in this area don't wait until a product is nearly finished to think about accessibility. Instead, they embed it into their workflows from the start, making it a shared responsibility across all teams.

Making accessibility a team responsibility

One of the biggest and most common obstacles to accessibility is the assumption that it's someone else's job. Designers often expect developers to handle it. Developers assume it's the responsibility of QA teams. QA teams feel like their job would be a lot easier if everyone else took a bit more ownership. Things get even worse when an accessibility champion is identified in the organization, because then, everyone else can just wash their hands of that responsibility. One thing is for sure: accessibility is a team effort where everyone has a role to play. And without clearly defined responsibilities, accessibility quickly falls through the cracks.

Worse still is the situation where a small, under-resourced accessibility team is tasked with "fixing" issues that stem from decisions made much earlier in the process. These teams often find themselves stuck between design, development, business, and QA, trying to advocate for accessibility while being pressured to sign off on products that aren't ready. They spend their time firefighting and justifying why barriers exist, while the rest of the organization blames them for slowing things down or not providing quick fixes.

It's an impossible position to be in, and one that almost always leads to burnout, frustration, and stalled progress. Ask anyone who's been there.

Picture a software company struggling with this exact issue. Their accessibility fixes are always coming in late, slowing down releases, frustrating developers, and leading to patchwork solutions instead of lasting improvements. By restructuring their design and development workflows so that accessibility is no longer treated as an afterthought, they can stop relying on their designated accessibility champion and instead start to embed accessibility tasks into every role for improved success.

Designers can then take charge of ensuring color contrasts meet WCAG standards, that obvious focus indicators are designed to support keyboard navigation, and that images include pertinent text alternatives. As we've seen in *Chapter 10*, when discussing inclusive design patterns and components, developers who leverage semantic HTML, build in seamless keyboard navigation systems, and test their pages with automated testing tools and assistive technology from the beginning can really push accessibility forward. QA testers can include both automated and manual accessibility checks in their standard test plans, with clear feedback loops so issues can be addressed in a timely manner.

By making accessibility a shared responsibility, the company eliminates many of the back-and-forth fixes and significant cost overruns that plague and slow down their projects. Within months, accessibility becomes a more natural part of their processes, rather than a hurdle to clear at the end.

Shifting left: building accessibility into the process from the start

The shift-left concept is simple: the earlier digital accessibility is considered as part of the process, the easier it is to integrate into the final product. When teams define accessibility requirements from the very beginning of a project, they avoid major accessibility barriers later on.

Imagine a company struggling with accessibility inconsistencies across its products. Some teams prioritize it, and others treat it as an afterthought. To fix this, they decide to build an accessible design system. Instead of asking every team to reinvent the wheel, they create shared components: accessible widgets, usable color palettes, legible typography, and well-tested interactive elements. With accessibility baked into the system, teams no longer have to guess. The burden on individual designers and developers drops. Accessibility becomes the default.

Another organization might take a different path. They build accessibility reviews into their sprint cycles, checking new features every two weeks instead of waiting until the end. Doing so helps the teams catch issues early while they're still easy to fix. Over time, this reduces last-minute scrambles and leads to more consistently accessible products.

Teams that adopt either approach are most successful when they also build in testing at every step. That means layering automated scans into each code commit to catch common issues such as missing alt text, unlabeled form fields, or contrast problems as soon as they appear. But they don't stop there. Before major releases, they also run usability tests with real users of assistive technology, surfacing deeper issues that automated tools can't detect, such as confusing keyboard focus order or broken screen reader interactions.

By testing early and often, teams catch more barriers before they go live and save time and effort in the long run. This approach to testing is what helps them go beyond technical compliance, so they can successfully create inclusive experiences for people living with disabilities.

Both strategies – the accessible design system and the sprint-based accessibility reviews – are valid ways to shift left. What really matters is that the accessibility practice is intentionally and consistently baked into the process. Whether you build it once at the foundation or revisit it regularly through agile cycles, the key is making it part of how you work, not something you scramble to fix later.

Reducing accessibility debt

Ignoring how accessibility issues impact end users also creates inclusion debt for organizations. It keeps adding to a backlog of issues that only grows harder and more costly to fix over time. Just like technical debt, the longer these problems go unaddressed, the more expensive and time-consuming they become to resolve.

Though he was never involved in the discipline, it is still Frank Lloyd Wright, an American architect, designer, writer, and educator who captured the essence of digital accessibility best when he famously said the following about the benefits of planning ahead of time when designing buildings: "*You can use an eraser on the drafting table or a sledgehammer on the construction site. Your choice.*"

This reminds me of a media company I once worked with that had been accumulating accessibility debt for years. Every new release introduced more barriers, and their backlog of accessibility fixes kept growing. They knew they needed a long-term solution, but all they could really come up with were temporary patches. To break the cycle, they formed an accessibility strike team: a dedicated group of designers, developers, and QA testers whose role was to review features before they were shipped. Within a year, they had cut their backlog by close to 60% and were finally releasing products that were that much more accessible from day one.

Putting together an accessibility strike team is just one of many approaches an organization can take to stay ahead of its accessibility debt. Other options include prioritizing accessibility fixes in backlog grooming, treating them like any other critical issue instead of putting them off for later. Other teams might choose to add accessibility checkpoints to their design reviews, so that potential barriers are caught before development even starts.

The most forward-thinking teams might even invest in training their designers and developers on accessible coding and design patterns, in order to reduce the chances of accessibility problems being introduced in the first place. In every one of these scenarios, proactivity is key.

No matter the strategy, what makes the biggest difference is building visibility and accountability into the process to prevent those issues from coming back. Without tracking what's been fixed and what still needs attention, it's far too easy for accessibility debt to start piling up again. And the best way to mitigate this risk is to treat accessibility issues like any other business priority: by tracking and reporting on them.

Measuring and reporting on accessibility progress

The reality is, your accessibility efforts won't mean much if your organization can't track its progress. This is because without clear benchmarks and reporting, accessibility tends to remain an abstract goal rather than a measurable outcome. Teams that make real progress are those that treat accessibility like any other business priority. They set goals, they create accountability, and they review results regularly.

Still, to this day, most organizations continue to assume that accessibility is either "done" or "not done." But that mindset is what prevents real progress... Accessibility is an ongoing practice with a lot (*and I mean a lot!*) of moving parts. One that can and does improve over time, given that the teams have the right data to guide their decisions. Measuring accessibility effectively means understanding the W3C guidelines, yes. But more importantly, it means understanding the actual experience of users with disabilities and tracking the long-term impact of the contributions and feedback.

Defining success beyond compliance

Many organizations start by measuring accessibility through compliance standards such as WCAG, the Americans with Disabilities Act (ADA), the Accessible Canada Act (ACA), the European Accessibility Act (EAA), or some other local, legal requirements. While compliance is important, it's not the finish line. In fact, compliance is nothing more than a good starting point. A product or service can absolutely meet technical standards on paper and still be frustrating, or even impossible for individuals with disabilities to navigate.

At this point, it would be fair to ask: if meeting WCAG doesn't guarantee a usable experience, what's the point of having these standards in the first place?

Now, don't get me wrong. WCAG still plays a critical role by providing the technical baseline that helps teams align their work and catch common issues, but it does not pretend to (nor was it ever designed to) capture everything about user experience or usability. What WCAG defines is the floor, not the ceiling. That's why teams that stop at accessibility compliance often fall short. Real accessibility requires moving beyond checklists and actively listening to how people interact with your products in the real world.

I've experienced this with clients more times than I can count over the past 25 years. As I write this, I'm reminded of a large e-commerce company that got to learn this firsthand. They had an internal accessibility audit process and were confident they were meeting WCAG standards. But when they brought in blind customers to test their site with various screen readers, they were surprised to find out that many users struggled to complete purchases.

Their navigation structure was technically compliant but functionally unusable. While most of the components used in their pages were, in isolation, meeting accessibility guidelines, the interface as a whole relied heavily on visual cues to tie everything together. Since many of those cues weren't conveyed properly to assistive technologies, it created serious comprehension gaps. Non-sighted users couldn't follow what was happening, and the experience broke down.

Realizing this, they started shifting their focus from compliance to usability and, based on the feedback received, redesigned their key interactions. The outcome was an overall better experience for all customers and a measurable increase in potential conversion rates.

In my experience, the organizations that move beyond compliance and take a more holistic view of accessibility are the ones who make the biggest strides. Instead of asking, "Are we compliant? Are we meeting the guidelines?", they start asking better questions:

- Can users with disabilities navigate, interact with, and complete tasks easily?
- Is the experience consistent across platforms, including mobile apps, websites, and documents?
- Are employees adopting accessibility practices daily, or is it still treated as an afterthought?

A simple shift in mindset – moving from compliance to usability – can go a long way.

Choosing the right accessibility metrics

Organizations can reliably measure accessibility success by tracking a mix of technical, usability, and process-related indicators. The most successful ones don't rely on a single number or key performance indicator (KPI) to tell the full story, such as how many WCAG success criteria their products happen to meet; they combine different data points to get a clear picture of where they're improving and where they need to focus next.

Metrics such as the percentage of pages passing automated scans, the average number of issues flagged per page, and the ratio of resolved to unresolved accessibility bugs in the backlog are all examples of useful metrics. Teams might also track the number of accessibility-related tickets submitted by users, the average time it takes to resolve those issues, or how often accessibility blockers delay product releases. On the process side, they might look at the percentage of team members who have completed accessibility training, the number of accessibility checkpoints integrated into design and development workflows, or how often assistive technology users are included in usability testing.

Imagine a healthcare company that wants to track its accessibility progress more systematically. To do this, it might introduce quarterly accessibility scorecards, review its automated test results carefully, capture and track user-reported issues, and gather feedback from assistive technology users.

In their case, this might include tracking the accessibility of digital appointment booking tools, the usability of electronic medical records for patients and providers, or how well their mobile app supports screen readers during prescription refills. Over time, they would likely notice patterns, such as certain teams making consistent improvements, while others keep on repeating the same accessibility mistakes over and over again.

I can't think of a better way for an organization to identify and fix accessibility issues across its digital assets than to target specific areas for training and improvement. Without such tracking, how could any team possibly identify what their weakest links are?

But for any accessibility metric to be meaningful, teams need insights that are both technical and human-driven. Common barriers can be reliably tracked with automated testing, but only real users have the ability to reveal just how accessible a product or service truly is based on lived experiences. To get a deeper understanding of their accessibility progress, it's in an organization's best interest to combine both.

Embedding accessibility into reporting

Also, it's not unusual for many leadership teams to struggle with keeping accessibility top of mind. This mistake typically results in accessibility fixes being constantly deprioritized, especially when teams are expected to focus on shipping new features or hitting tight deadlines.

One of the key lessons I learned over the years is that accessibility should never be treated in parallel, or as a separate initiative, but instead, built directly into how organizations measure success. When accessibility reporting is woven into existing business reviews, when accessibility considerations are baked right into existing processes, it becomes that much harder for everyone to ignore. Accessibility requirements stay visible, and they get the attention they deserve.

This can be achieved by integrating accessibility data into executive dashboards, alongside other important business metrics such as security, performance, and user satisfaction. Doing so then ensures that accessibility successfully moves from an obscure discussion in specialized meetings to a regular topic in leadership reviews.

Taking this approach also has the benefit of setting clear accessibility performance targets. Some teams might track progress by aiming to reduce accessibility errors by a certain percentage within six months, while others will prefer to focus on increasing user satisfaction scores from people with disabilities. Whatever the metrics, teams that have specific, measurable goals are more likely to prioritize accessibility in their daily work.

Building transparency and accountability

But it doesn't stop there. For accessibility efforts to succeed long-term, progress must be visible and celebrated regularly. When organizations keep accessibility results transparent, teams take it more seriously, leadership stays engaged, and progress is easier to sustain.

A nice way to do this is to introduce a monthly accessibility leaderboard, where each team might be ranked based on their accessibility improvements. This isn't about competition, though competition might be a strong motivator in a lot of cases. This is about visibility. Teams that make accessibility improvements are recognized internally, and those falling behind can see where and how they need to improve. With the right framing from leadership, this friendly accountability can encourage teams to fix issues faster and take ownership of accessibility.

Some organizations choose to build transparency by publicly sharing their accessibility commitments. I'm a big fan of this approach as I believe it reinforces commitment and accountability. Though some are forced to publish accessibility progress updates in their annual reports, some might choose to do it spontaneously, while others might prefer to issue statements on their websites, outlining what they've improved and what they're still working on.

There's more than one way to build transparency, of course, but all of those approaches to openness ultimately contribute to building trust. And not just with customers, but internally with the workforce as well. Employees take accessibility much more seriously when they can see that the organization is holding itself accountable as well.

Sustaining progress over time

While the goal for most organizations is digital inclusion and legal compliance, measuring accessibility shouldn't just be about proving that the work is done. Measuring should be part of a larger commitment to broadening the reach of the brand because the organization understands that better servicing their target audiences through the sustained long-term improvement of digital properties is the smart business decision to make.

Now, it's fair to ask: if accessibility is legally mandated and often shaped around standards such as WCAG, why go beyond that? Why should organizations aim higher when meeting the minimum legal threshold if this is all that is technically required? After all, no one ever asks companies to go above and beyond the law in other areas…

The difference is that accessibility isn't just about rules and regulations; it's about people. Accessibility compliance will keep you out of legal trouble, but compliance alone won't help you build the trust or earn the loyalty of your disabled audiences, and it certainly won't allow you to fully tap into their communities either.

One thing is for sure: organizations that understand how accessibility brings much sought-after values, such as better customer experiences, broader reach, and stronger brand integrity, very naturally begin to treat accessibility as a human issue more than a legal one.

You'll hear many experts say that accessibility is just as much about the journey as it is about the destination. What this means is that successful organizations don't just set one-time accessibility goals for themselves; they recognize accessibility as an ongoing commitment and build it into their ongoing review processes, so that their teams can keep improving year after year.

The teams that go beyond checkbox compliance, track meaningful usability metrics, and make accessibility a visible part of business reporting are the ones who truly shift culture. They prioritize people and deliver better customer experiences as a result. And that, my friends, is good for the bottom line, too.

Therefore, beyond tracking progress or embedding it in reports, sustaining accessibility is primarily a question of building the kind of culture where learning never stops. Standards evolve. Tools change. User expectations grow. The organizations that stay ahead are the ones that treat accessibility not as a fixed destination but as a skillset to develop, a mindset to nurture, and a shared responsibility that grows stronger through continuous learning.

Weaving continuous learning into inclusive design

It really goes without saying. Accessibility isn't a one-time fix or a project with a set start and end date; it's an ongoing, continuous practice that evolves alongside technology, user needs, business goals, and industry standards.

The proof is in the pudding: organizations that commit to ongoing accessibility learning are also the ones that build stronger teams and develop more inclusive products and services. But accessibility education is only effective when it's practical, engaging, and woven into the daily work of every stakeholder.

Way too many teams treat accessibility training as a single event or one-off sessions filled with guidelines and compliance requirements. Having spent the last 15+ years of my career as a corporate accessibility trainer, I can assure you that I speak from experience here. Sadly, information alone won't change deeply-rooted behaviours.

The organizations that succeed the most are always the ones that make accessibility a part of their culture and make sure that learning never stops!

Making accessibility training meaningful

In my opinion, one of the biggest mistakes organizations make is delivering accessibility training in a way that feels disconnected from real, tangible work. Long presentations filled with lectures about regulations, technical rules, and best practices often fail to stick and do very little in terms of generating engagement and retention. Accessibility is most impactful when it's anchored in concrete concepts that participants can relate to, and tailored to specific roles in the life cycle that give teams many opportunities to apply what they learn immediately.

Examples of companies that struggle with accessibility issues despite hosting annual training sessions are a dime a dozen. Employees, especially the ones who have been around for a while, will understand the importance of accessibility, but when deadlines get tight, accessibility slips through the cracks.

To fix this, organizations would be much better off replacing their mandatory annual training with monthly accessibility challenges. Instead of unengaging lecture-style sessions, teams could work on real accessibility barriers or issues in their projects, such as reviewing poor color contrast, improving keyboard navigation, or fixing screen reader problems.

Using such a hands-on approach precipitates improvements and reduces accessibility bugs in new releases by a significant margin. It also leads stakeholders to think about accessibility proactively rather than waiting for issues to be flagged during audits. The shift from passive training to hands-on learning would make accessibility a natural part of their workflow rather than an occasional compliance task.

Encouraging peer learning and knowledge sharing

While formal training is very valuable, some of the best learning opportunities take place through day-to-day conversations and shared experiences, not by passively listening to theoretical lectures in classrooms (virtual or otherwise). When accessibility knowledge is distributed evenly across team members, and every role in the life cycle has a clearly defined set of responsibilities assigned to it, everyone feels more confident in asking questions, sharing solutions, and applying what they learn. And all of that leads to success.

One such example of a strategy could be to launch an internal accessibility mentorship program. Experienced accessibility advocates within the organization could be paired with team members who want to improve their skills in this area. The mentorship's influence could extend beyond the typical design and development groups, creating a wider reach of expertise. Internal accessibility supporters could naturally evolve into brand ambassadors, promoting accessibility best practices and making sure discussions about accessibility become commonplace across all company divisions.

Other groups might choose to create internal "safe spaces" for accessibility conversations, whether through dedicated Slack channels, discussion groups, or team meeting check-ins. When everyone involved, from designers, developers, testers, and every other stakeholder, has opportunities to collaborate on accessibility challenges, what they build is a shared understanding of best practices. They can then learn from each other's experiences, and all of that, again, leads to even more accessibility successes.

Keeping up with evolving standards and tools

Accessibility is always changing. New technologies, guidelines, and industry trends are constantly emerging. Teams that fail to stay informed run the risk of falling behind. The most effective organizations make ongoing accessibility education a priority to ensure that their teams remain up to date.

A client I worked with recently recognized this challenge early on. To keep accessibility knowledge current within their company, we launched a quarterly accessibility newsletter to demystify obscure guidelines, highlight industry updates, showcase internal success stories, and introduce new tools available. By sharing accessibility insights in a consistent manner, they make sure their teams don't lose momentum, that they don't fall back on outdated practices, and that accessibility remains top of mind.

Other organizations might take more structured approaches by offering incentives for employees to pursue accessibility certifications or by encouraging attendance at accessibility conferences and webinars. Events such as *AccessU*, *Inclusive Design 24*, *CSUN*, *a11yToronto*, and *Axe-Con* provide valuable learning opportunities at next to no cost, while also exposing teams to emerging best practices and innovations from top-level experts in the field.

Embedding accessibility into career growth

One of the best ways to sustain accessibility efforts is to tie accessibility skills to professional or career development. When accessibility is recognized by top-level executives and leadership as a valuable professional skill, employees are more likely to prioritize it in their work and develop the skill set.

Picture an organization that decides to launch an "Accessibility Excellence" award in order to recognize employees who make significant contributions to accessibility within their projects. Can you see how celebrating these efforts would reinforce the idea that accessibility is a key part of recognized professional growth within the company, as opposed to an item to check off a checklist?

Other organizations might choose to make accessibility a formal part of job descriptions, ensuring in the process that new hires already value inclusion when they join the team. Some companies might even intentionally include accessibility contributions as part of their teams' annual performance reviews, consequently making accessibility grow from a team goal to an individual responsibility.

When employees see that accessibility expertise leads to further recognition, leadership opportunities, and career growth, then the likelihood of them becoming more invested in developing and applying these skills long-term increases exponentially.

Creating a culture of continuous learning

The most inclusive organizations that I've come across are the ones that never stop learning and are always eager to question what they think they know about digital accessibility. They're the ones who stopped treating accessibility as a project or a checklist to complete, and instead, made it a constant and continuous part of their growth and innovation.

Time and time again, I've seen how teams that focus on hands-on, role-specific training build deeper accessibility knowledge than those that only focus on learning about the "rules." Leaders who encourage peer learning and internal mentorship also end up creating stronger, more inclusive, and more confident teams.

Companies that proactively involve end users with disabilities in their process and make it a point to keep up with evolving standards are the ones who have an easier time preventing accessibility gaps before they become problems. And clearly, when accessibility skills are recognized as a valuable part of career development, employees can see what's in it for them. As a result, they are also much more likely to stay engaged and committed to the organization's long-term accessibility success.

The goal should always be for accessibility education to be practical, engaging, and continuous, so it stops being perceived as just another pointless requirement, and instead, becomes a gratifying, shared responsibility that ultimately drives better, more inclusive products and services for everyone involved.

Summary

If there's one key message I hope you take away from this chapter, it's that building an inclusive design culture is about more than checking boxes or hitting compliance targets. Sure, those things matter because they're often how your efforts will be measured, but the real value goes much deeper than that. It's about bringing accessibility into the very DNA of how teams work, day in and day out, across every role and at every level. When that shift happens, everything changes. And so does your culture.

As we've explored in this final chapter, it all starts with building empathy. Not the surface-level kind, but the kind that comes from truly listening, learning, and connecting with people who use assistive technologies and experience the web differently. Real accessibility comes from true understanding. Inclusive design starts when we shift our focus from general ideas to the actual people who will use what we create.

This is the moment where the path forward becomes clearer. When accessibility is integrated into workflows – from project planning through design, development, and testing – it stops being a scramble at the finish line and becomes something teams can focus on from the start. It becomes part of how good, inclusive design work gets done!

Of course, none of that sticks without accountability. The teams that grow are the ones that track their progress, share results openly, and let data guide their next move. Instead of simply following the rules, they ask probing questions such as "Is this beneficial?" and pay close attention to what people say.

And because the field is always changing, so should our approach! The most inclusive organizations are the ones that keep learning, invest in hands-on training, encourage knowledge sharing, and make space for accessibility to grow. They recognize effort, reward progress, and treat accessibility as a skill worth developing, not just a checklist item to complete before moving on to the next project.

So, if you're wondering what to do next, don't overthink it. Start small. Choose one thing. Run a quick usability test with someone who uses assistive technology. Review your design system for color contrast issues, text alternatives to images, or keyboard navigation. Talk to your team about making accessibility part of sprint planning. Whatever it is, just start.

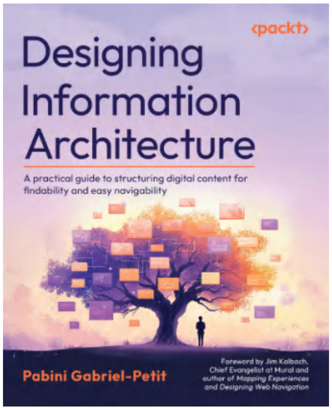
Because the truth is, inclusive design isn't a destination, it's a habit. A mindset. A shared commitment that gets stronger with every action you take. And the more you do it, the more natural it becomes. Commit, and soon, you'll start to see the difference for yourself. Not just in your products and services, but in your people, your process, and your purpose.

Now, go ahead. Take that first step and watch your future users thank you for it. And honestly? So will your team because, in the end, we all care about creating amazing, meaningful products and services. Great design should never be just about *what* works. It should also be about *who* it works for.

And that's where the magic of inclusive design really happens, if you ask me.

Other Books You May Enjoy

If you enjoyed this book, you may be interested in these other books by Packt:



Designing Information Architecture

Pabini Gabriel-Petit

ISBN: 978-1-83882-390-0

- Information-seeking models, strategies, tactics, and behaviors
- Principles for designing IAs that support human cognitive and visual capabilities
- Wayfinding principles for placemaking, orientation, navigation, labeling, and search
- Useful structural patterns and information-organization schemes
- UX research methods and analytics for information architecture
- Content analysis, modeling, and mapping methods
- Categorizing content and creating controlled vocabularies
- Designing and mapping information architectures
- Leveraging artificial intelligence (AI) to deliver optimal search results

101 UX Principles – 2nd edition

Will Grant

ISBN: 978-1-80323-051-1

- Work with user expectations, not against them
- Make interactive elements obvious and discoverable
- Optimize your interface for mobile
- Streamline creating and entering passwords
- Use animation with care in user interfaces
- How to handle destructive user actions

Packt is searching for authors like you

If you're interested in becoming an author for Packt, please visit `authors.packtpub.com` and apply today. We have worked with thousands of developers and tech professionals, just like you, to help them share their insight with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Share Your Thoughts

Now you've finished *Inclusive Design for Accessibility*, we'd love to hear your thoughts! Scan the QR code below to go straight to the Amazon review page for this book and share your feedback or leave a review on the site that you purchased it from.

`https://packt.link/r/1-835-88823-2`

Your review is important to us and the tech community and will help us make sure we're delivering excellent quality content.

Index

A

Acceptance Criteria (AC) 253

accessibility

barriers 258, 259

building, into process from start 265

debt, reducing 265, 266

embedding, into reporting 269

integrating, into workflows 263

making, team responsibility 263, 264

progress, sustaining over time 270, 271

transparency and accountability, building 270

accessibility progress

measuring 266

metrics, selecting 268, 269

reporting 266

success, defining beyond compliance 267

Accessibility Roles and Responsibilities Model (ARRM) 143

accessibility strike team 266

accessibility testing

integrating, into development workflows 252-254

accessibility test results

bugs, prioritizing 249

dashboards 251, 252

interpreting 249

prioritizing 249

Accessible Canada Act (ACA) 267

accessible design

ethical dimensions 52

accessible design system patterns

for cognitive disabilities 220

for color perception 218

for deaf or hard of hearing 219

for low vision or blindness 229-232

for motion sensitivity 223

for motor disabilities 222-226

for photosensitivity 224

for speech disabilities 225

Accessible Rich Internet Applications (ARIA)

applying, to enhance dynamic and custom components 160

aria-expanded attribute 158, 159

aria-labelledby 159

dynamics, enhancing with 156, 157

rules 158

AI governance
accessibility, addressing 57, 58
AI-powered predictive text for users with motor disabilities 200-202
algorithmic bias, in AI-driven interfaces
ableism and unconscious biases 203, 204
best practices 204, 205
data bias and representation 203
FairFace 206
mitigating 202
alternative input methods
Android keyboard 177
custom solutions 176
focus order, in mobile accessibility 178, 179
implementing 175
iOS keyboard 178
platform controls 176
users usability 180, 181
alternative text (alt text) 246
Amazon Alexa
reference link 108
Americans with Disabilities Act (ADA) 45, 267
Android accessibility
versus iOS accessibility 166, 167
AR accessibility, via customization 196
AR experiences, designing 196
AR overlays 197, 198
AR experiences
auditory accommodations 197
cognitive accommodations 197
designing 196
mobility accommodations 197
visual accommodations 197
aria-expanded attribute 158
aria-labelledby attribute 159
Artificial Intelligence (AI) 192
Artificial Intelligence (AI) accessibility practices
algorithmic bias 200
building 200
equitable AI design 200
predictive text 200
using 199
assistive technologies
for auditory disabilities 30
for cognitive disabilities 33
for motor disabilities 32
for visual disabilities 29
impact, on design 35
integrating, into design 34
role, examining 28
types 28, 29
assistive technologies, for auditory disabilities 30
captions 31
hearing aids and cochlear implants 31, 32
transcripts 31
visual alerts 31
assistive technologies, for cognitive disabilities 33
readability enhancements 33, 34
simplified interfaces 33, 34
speech-to-text tools 33
text-to-speech tools 33
assistive technologies, for motor disabilities
adaptive input devices 32
keyboard navigation 33
voice recognition 32
assistive technologies, for visual disabilities
audio description 30
Braille displays 30
disabilities 30
high-contrast tools 30
screen readers 29

audio descriptions 30

auditory disabilities 26

Augmented Reality (AR) 192, 196

automated solutions
- role and limitations 51, 52

automated testing tools
- selecting 240-242

B

bugs, prioritizing
- blockers and serious issues 249
- factor considerations 250, 251
- moderate issues 250

business case, for inclusivity
- brand reputation, enhancing 19
- innovation 19
- legal compliance and risk mitigation 20
- market reach, expanding 18
- recognizing 17, 18

C

captions 31

card pattern 170, 171

carousel slideshows 236

Cascading Style Sheets (CSS)
- used, for avoiding addition of important content 151, 152
- used, for styling inclusive design 150

case studies 136-138

Cephable
- reference link 108

chunking 121

code testing
- for accessibility conformance 160-162

cognitive disabilities 27
- design system 220, 221

color
- used, for conveying errors 100, 101
- used, for denoting action and interaction 101, 102
- using, for time-sensitive decision-making and safety 96, 97
- using, in charts and diagram 99, 100
- using, in visual design 96

color contrast
- using 98, 99

Color Contrast Checker 99

color perception
- design system 218, 219

color vision deficiency 218

company-wide commitment
- driving 262

continuous learning
- accessibility, embedding into career growth 273
- accessibility training, making meaningful 272
- culture, creating 274
- evolving standards and tools 273
- knowledge sharing, encouraging 272
- peer learning, encouraging 272
- weaving, into inclusive design 271

Convention on the Rights of Persons with Disabilities (CRPD) 45

D

Definition of Done (DoD) 253

designing, for accessibility 109
- accessible design system, building 110
- design considerations, for elements 110, 111
- information architecture, for accessibility 112
- personalization, empowering through user customization 111

design solutions
gaps, bridging 41
gaps identification 41
iteration 40
user journey 36
user needs, mapping to 35, 36
design system audio and video
closed captions 219
transcripts 220
design system, for low vision 212
affordances, for magnification tools 217, 218
dynamic typography, planning for 213
typographic patterns, delivering 213-217
web browser zoom, enabling 213
digital accessibility 48, 49
designing for disabilities, to human diversity 5
diverse spectrum of user needs and abilities 6
future legal trends, forecasting 54-58
historical context and evolution 4
legal foundation, establishing 47, 48
roadblocks, spotting 7
social model disability
digital contexts
auditory disabilities 26
cognitive disabilities 27
disabilities types, categorizing 25
motor disabilities 27
visual disabilities 25
digital services
reliance, increasing on 54
disability simulations
reasoning, for falling short 260

E

empathy
building, through real engagement 261
cultivating, through immersive experiences 258
embedding 261
turning, into action 262, 263
ethical dimensions
of accessible design 52
ethically driven design culture
cultivating 58-60
European Accessibility Act (EAA) 48, 267
explainable AI (XAI) 206

F

FairFace 206
floating action buttons 172, 173
best practices 173
layout 173
platform standards 173
user expectation 173
focus order
in mobile accessibility 179
techniques 179, 180

G

global accessibility legislation
challenges 50
legal foundation, establishing for digital accessibility 47, 48
navigating 46, 47
global legal frameworks 48, 49
Grammarly
URL 127

H

Hypertext Markup Language (HTML) 146

I

images 127
alt text, utilizing 128, 129
format combinations 132
graphs and charts 129
icons 130
text, avoiding 131
types 127, 128
inclusive design principles
implementing 9-11
inclusive design solutions
research insights, translating into 86-90
inclusive design, styling with CSS 150, 156
color contrast 150, 151
content addition, avoiding 151, 152
design, for flexibility and scale 153
user preferences 152, 153
visible focus styles, providing 154
inclusive research 65
designing, from edges 69, 70
disabled lives, centering through reciprocal recruitment 71, 72
diverse participants, recruiting 68
implementing 76, 77
recruitment, as relational practice 68
inclusive writing practice
building, key actions 117, 118
interactive voice response (IVR) 225
International Organization for Standardization (ISO) 97
intersectionality
designing for 234-237
iOS accessibility
versus Android accessibility 166

J

Job Access With Speech (JAWS) 244

K

keyboard navigation 33
key performance indicator (KPI) 268

L

legal compliance and ethical responsibility
balancing 51
role and limitations, of automated solutions 51, 52
lived user experiences
culture of accountability, creating 43
feedback, integration into workflows 42
insight, building through 41, 42

M

manual accessibility audits
checks 247-249
color vision deficiency 247
image description, checking 246
keyboard navigation 243, 244
page, testing 246, 247
performing 242
screen reader 244, 245
screen reader, tips 246
zoom in 247
mobile context 179
mobile development accessibility considerations 166
Android accessibility 166
iOS accessibility 166
maintainability 167, 169
motion sensitivity
design system 223

motor disabilities 27
design system 226, 228
multiple alternative input methods
assistive technologies 109
designing 106
gesture controls 108
voice commands 107

N

Needs Walkthrough exercise
conducting 138, 141
content navigation, with WCAG 139
disability need list, gathering 138, 139
exercise event, planning 141
issues, solving 143
learnings 142
standards and outcomes, identifying 139, 140
non-semantic HTML
defining 147
NonVisual Desktop Access (NVDA) 244

P

Photosensitive Epilepsy Analysis Tool (PEAT)
reference link 224
photosensitivity 224
design system 224
plain language
URL 118

Q

qualitative data
analyzing, from diverse user groups 80
dynamic data 84-86
meaning-making, responsibility 82-84
neurodivergence, as method 81

R

real-world applications, mapping needs 37
banking and financial tools 39, 40
e-commerce websites 38
e-learning platforms 37
refreshable Braille displays (RBDs) 29
reporting
accessibility, embedding into 269
research methods
adapting, for accessibility 73
dynamic and human-centered practice, research as 79
empowered research spaces 74, 75
flexible communication methods, providing 77, 78
inclusive research, implementing 76, 77
lived experience 75, 76
loop, closing with participants 78
responsive layouts
adaptive layouts 187
ensuring 185
information density 186
standard tools utilization 187
role 245

S

screen magnifiers 30
semantic HTML
defining 148
used, for customizing websites 149
used, for structuring content 146, 147
shift-left concept 264
social media 135, 136
Software Development Life Cycle (SDLC) 252

spectrum
disability 24, 25
speech disabilities 226
design system 225
speech-to-text (STT) 33
success criteria (SC) 143

T

Telecoil (T-coil) systems) 31
text
bulleted and numbered lists 122
form field labels and instructions 123
group related form inputs under headings 124
headings 121
labels, for buttons and interactive controls 123
links 123
making accessible 120
making accessible, through scannability 120, 121
navigation landmarks and menu links 122
navigation menu and button labels 122
paragraphs 121
plain language principles 125-127
text-to-speech (TTS) 33
text version 136
tick-box exercise 53, 54
touchable area extension 174
touch targets
card pattern 170, 171
floating action buttons 172, 173
optimizing 169, 170
touchable area extension 174
two-dimensional scrolling 171, 172
workflows, utilizing 174, 175
transcripts 31
two-dimensional scrolling 171, 172

U

universal design
principles 8, 9
user-centered design methodologies
example, Microsoft's Xbox Adaptive Controller 15, 16
implementing 12
inclusive user research and testing techniques 14
iterative design processes 15
user needs and experiences 12, 13
user interfaces, designing for cognitive disabilities 102
consistency 102-104
progressive disclosure 104
reversal of actions 105
unnecessary content abstraction 104, 105

V

video and audio 132
closed captions 132, 133
format combinations 134
text transcripts 133, 134
Virtual Reality (VR) 192
visual design
color, using 96
visual disabilities 25
voice interfaces, for accessibility
designing 181-183
identify controls 183
nesting controls, avoiding 184

VR accessibility
auditory accommodations 193
cognitive accommodations 193
considerations 194
improving 193
mobility accommodations 193
motion sickness mitigation 194
visual accommodations 193
VR experiences, for sensory abilities 192
virtual escape room 194, 195
VR accessibility, improving 193

W

WCAG standard principles
operability 119
perception 119
robust 120
understanding 119
Web Accessibility Initiative (WAI) 117
Web Content Accessibility Guidelines (WCAG) 48, 98, 117, 150, 261
web context 179
World Wide Web Consortium (W3C) 117, 156

Download a Free PDF Copy of This Book

Thanks for purchasing this book!

Do you like to read on the go but are unable to carry your print books everywhere?

Is your eBook purchase not compatible with the device of your choice?

Don't worry, now with every Packt book you get a DRM-free PDF version of that book at no cost.

Read anywhere, any place, on any device. Search, copy, and paste code from your favorite technical books directly into your application.

The perks don't stop there, you can get exclusive access to discounts, newsletters, and great free content in your inbox daily.

Follow these simple steps to get the benefits:

1. Scan the QR code or visit the link below:

   `https://packt.link/free-ebook/9781835888223`

2. Submit your proof of purchase.
3. That's it! We'll send your free PDF and other benefits to your email directly.