Digital Accessibility Testing (CT-DAT) Syllabus

Version 1.0

International Software Testing Qualifications Board



© International Software Testing Qualifications Board

Copyright Notice

Copyright Notice © International Software Testing Qualifications Board (hereinafter called ISTQB®)

ISTQB® is a registered trademark of the International Software Testing Qualifications Board.

Copyright © 2026 the authors of the Digital Accessibility Testing v1.0 syllabus: Graham Bath (WG Chair), Andrew Pollner (WG Vice Chair), Rimvydas Petkus, Sophie van der Sijp, Kimberly Snoyl, Gary Mogyorodi

All rights reserved. The authors hereby transfer the copyright to the ISTQB®. The authors (as current copyright holders) and ISTQB® (as the future copyright holder) have agreed to the following conditions of use:

Extracts, for non-commercial use, from this document may be copied if the source is acknowledged. Any Accredited Training Provider may use this syllabus as the basis for a training course if the authors and the ISTQB® are acknowledged as the source and copyright owners of the syllabus and provided that any advertisement of such a training course may mention the syllabus only after official Accreditation of the training materials has been received from an ISTQB®-recognized Member Board.

Any individual or group of individuals may use this syllabus as the basis for articles and books, if the authors and the ISTQB® are acknowledged as the source and copyright owners of the syllabus.

Any other use of this syllabus is prohibited without first obtaining the approval in writing of the ISTQB®.

Any ISTQB®-recognized Member Board may translate this syllabus provided they reproduce the Copyright Notice mentioned above in the translated version of the syllabus.

Revision History

Version	Date	Remarks
v1.0 Alpha	2026/04/27	ISTQB Alpha Release
v1.0 Beta	2026/06/22	ISTQB Beta Release
v1.0	2026/10/30	ISTQB GA Release

© International Software Testing Qualifications Board

Table of Contents

0. Introduction 8
0.1 Purpose of this Syllabus 8
0.2 The Digital Accessibility Testing Qualification 8
0.3 Career Path for Testers 8
0.4 Business Outcomes 8
0.5 Learning Objectives and Cognitive Level of Knowledge 9
0.6 The Digital Accessibility Testing Certificate Exam 9
0.7 Accreditation 10
0.8 Handling of Standards 10
0.9 Level of Detail 10
0.10 How this Syllabus is Organized 11
1. Basic Concepts – 90 minutes 13
1.1 Fundamentals of Digital Accessibility 13
1.1.1 Digital Accessibility 13
1.1.2 The Objectives of a Digital Accessibility Evaluation 14
1.1.3 Principles and Practices for Digital Accessibility Testing 15
1.2 Disabilities 16
1.2.1 Categories of Disabilities 16
1.2.2 The Importance of Accessible Software 18
1.3 Risks of Inadequate Digital Accessibility 19
1.3.1 Risks and Impact of Not Applying Digital Accessibility to Software 19
2. Digital Accessibility Standards – 75 minutes 21
2.1 Digital Accessibility Guidelines 21
2.2 Digital Accessibility Regulations and Standards 22
2.2.1 Standards and Legislation 22
2.2.2 General content and applicability of ISO 9241-171:2025 23
2.2.3 The Web Content Accessibility Guidelines 24
3. Digital Accessibility Testing Activities – 120 minutes 27
3.1 The Approach to Digital Accessibility Testing 28
3.1.1 Planning Activities for Digital Accessibility Testing 28

© International Software Testing Qualifications Board

3.2 Digital Accessibility Testing Activities 29
3.2.1 Benefits of Shift Left in Digital Accessibility Testing 29
3.2.2 How to Perform Digital Accessibility Testing 30
3.2.3 Digital Accessibility Audits as a Method for Evaluating Conformance 31
3.2.4 The Relevance of End-user Testing in Achieving a Highly Functional Product 32
3.3 A Step-By-Step Approach to End-User Testing 33
3.3.1 The Purpose and Approach Adopted in End-user Testing 33
3.4 Risks and Defects in Digital Accessibility Testing 34
3.4.1 Typical Project Risks and Mitigation Measures 34
3.4.2 Common Software Defects in Digital Accessibility Testing 35
4. Digital Accessibility Testing Lifecycle – 210 minutes 37
4.1 Preparing a Digital Accessibility Test 37
4.1.1 Factors to Consider in Digital Accessibility Test Planning 37
4.1.2 Configuration Requirements for Digital Accessibility Tests 39
4.2 Executing Digital Accessibility Tests 40
4.2.1 Identifying Applicable Digital Accessibility Test Tasks 41
4.3 Analyzing and Communicating Findings 42
4.3.1 Analyzing the Digital Accessibility Test Results 42
4.3.2 Typical Criteria Associated with Findings 43
4.3.3 Best Practices in Reporting and Communicating Test Results 44
4.3.4 The Purpose and Structure of a Formal Digital Accessibility Report 46
5. Assistive Technology – 75 minutes 48
5.1 Introduction to Assistive Technology 48
5.1.1 Understanding Assistive Technologies 48
5.1.2 Using Assistive Technologies 50
6. Tools for Digital Accessibility Testing – 105 minutes 53
6.1 Types of Automation Tools for Digital Accessibility Testing 53
6.1.1 Test Automation and its Limitations in Digital Accessibility Testing 53
6.1.2 The Role and Risks of Artificial Intelligence in Digital Accessibility Testing 55
6.2 Tools for Digital Accessibility Testing 56
6.2.1 Categories of Tools Available 56
6.2.2 Category 1: Static Analysis and CI/CD Pipeline Tools 59
6.2.3 Category 2: Automated UI-based Scanners and Manual Helpers 60

© International Software Testing Qualifications Board

6.2.4 Category 3: Assistive Technologies 61
6.2.5 Category 4: Cognitive and Readability Tools 62
7. References 65
7.1 Standards 65
7.2 ISTQB Documents 65
7.3 Books 65
7.4 Other References 65
8. Appendix A – Learning Objectives/Cognitive Level of Knowledge 67
9. Appendix B – Business Outcomes Traceability Matrix with Learning Objectives 70
10. Appendix C – Release Notes 78
11. Appendix D – Domain-Specific Keywords 79
12. Index 82

© International Software Testing Qualifications Board

Acknowledgements

This document was formally released by the General Assembly of the ISTQB® on <date>

It was produced by a team from the International Software Testing Qualifications Board: Graham Bath (WG Chair), Andrew Pollner (WG Vice Chair), Rimvydas Petkus (Product Owner & Author), Sophie van der Sijp (Key Author), Kimberly Snoyl (Author).

Technical Reviewer: Mike Smith

The following persons participated in the reviewing, commenting and balloting of this syllabus:

Gery Agnecz, Laura Albert, Debbie Archer, Anuradha Biswas, Armin Born, Earl Burba, Jean-Baptiste Crouigneau, Taz Daughtrey, Florian Fieber, Ray Gillespie, Agota Horvath, Leanne Howard, Corne Kruger, Marek Majerník, Aneta Masárová, Judy McKay, Korkor Mensah, Imola Mészár, Gary Mogyorodi, Frank Neirynck, Mwangi Samuel Njihia, Kenji Onoshi, Tauhida Parveen, Nishan Portoyan, Meile Posthuma, Annelies van Rijn, Dr. Irmhild Rogalla, Lea Schmitt, Klaus Skafte, Mike Smith, Michael Stahl, Lucjan Stapp, Ioannis Symeonidis, Richard Taylor, Nilo Thomas, Geoff Thompson, Julia Trinnes, M.N.A. van den Berg, Linda Vreeswijk

International Software Testing Qualifications Board Working Group Specialist Quality Characteristics and Test Levels (Edition 2026).

© International Software Testing Qualifications Board

0. Introduction

0.1 Purpose of this Syllabus

This syllabus forms the basis for the International Software Testing Qualification for Digital Accessibility Testing at Specialist level. The ISTQB® provides this syllabus as follows:

1. To member boards, to translate into their local language and to accredit training providers. Member boards may adapt the syllabus to their particular language needs and modify the references to adapt to their local publications.
2. To certification bodies, to derive examination questions in their local language adapted to the learning objectives for this syllabus.
3. To training providers, to produce training materials and determine appropriate teaching methods.
4. To certification candidates, to prepare for the certification exam (either as part of a training course or independently).
5. To the international software and systems engineering community, to advance software and systems testing profession, and as a basis for books and articles.

0.2 The Digital Accessibility Testing Qualification

The Digital Accessibility Testing qualification is aimed at anyone involved in software testing who wishes to broaden their knowledge of digital accessibility testing or anyone who wishes to start a specialist career in digital accessibility testing. The qualification is also aimed at anyone involved in software development who wishes to understand digital accessibility testing.

0.3 Career Path for Testers

The ISTQB® scheme supports testing professionals at all stages of their careers offering both breadth and depth of knowledge. Individuals who achieve the ISTQB® Digital Accessibility Testing certification may also be interested in the Usability Testing certification. [ISTQB_CT_UT]

0.4 Business Outcomes

This section lists the Business Outcomes expected of a candidate who has achieved the Digital Accessibility Testing certification.

A Digital Accessibility Testing Certified Tester can…

© International Software Testing Qualifications Board

DAT-BO1	Explain the value of digital accessibility principles and their impact on users.
DAT-BO2	Interpret global guidelines to apply digital accessibility standards.
DAT-BO3	Define digital accessibility testing activities
DAT-BO4	Identify, analyze, and report digital accessibility defects
DAT-BO5	Use assistive technology for digital accessibility testing
DAT-BO6	Select and use tools for digital accessibility testing

0.5 Learning Objectives and Cognitive Level of Knowledge

Learning Objectives support the business outcomes and are used to create the Digital Accessibility Testing exams.

In general, all contents of this syllabus are examinable, except for the Introduction and Appendices. The exam questions will confirm knowledge of keywords at K1 level (see below) or learning objectives at the respective level of knowledge.

The specific learning objectives and their levels of knowledge are shown at the beginning of each chapter, and classified as follows:

- K1: Remember
- K2: Understand
- K3: Apply
- K4: Analyze

Further details and examples of learning objectives are given in Appendix A.

For all terms listed as keywords just below chapter headings, the correct name and definition from the ISTQB® glossary shall be remembered (K1), even if not explicitly mentioned in the learning objective.

0.6 The Digital Accessibility Testing Certificate Exam

The Digital Accessibility Testing Certificate exam will be based on this syllabus. Answers to exam questions may require the use of material based on more than one section of this syllabus. All sections of the syllabus are examinable, except for the Introduction and Appendices. Standards and books are included as references, but their content is not examinable, beyond what is summarized in the syllabus itself from such standards and books.

Refer to Exam Structures and Rules document for the Digital Accessibility Testing module for further details.

The entry criterion for taking the Digital Accessibility Testing Specialist Level exam is that candidates are interested in digital accessibility However, it is strongly recommended that candidates also:

© International Software Testing Qualifications Board

- Have at least a minimal background in either software development or software testing, such as six months' experience as a system or user acceptance tester or as a software developer
- Take a course that has been accredited to ISTQB® standards (by one of the ISTQB-recognized member boards).

Entry Requirement Note: The ISTQB® Foundation Level certificate shall be obtained before taking the Digital Accessibility Testing certification exam.

0.7 Accreditation

An ISTQB® Member Board may accredit training providers whose course material follows this syllabus. Training providers should obtain accreditation guidelines from the Member Board or body that performs the accreditation. An accredited course is recognized as conforming to this syllabus and is allowed to have an ISTQB® exam as part of the course.

The accreditation guidelines for this syllabus follow the general Accreditation Guidelines published by the Processes Management and Compliance Working Group.

0.8 Handling of Standards

International standardization organizations like IEEE and ISO have issued standards associated with quality characteristics and software testing. Such standards are referenced in this syllabus. The purpose of these references is to provide a framework (as in the references to ISO 25010 regarding quality characteristics) or to provide a source of additional information if desired by the reader. Please note that ISTQB® syllabi are using the standards documents as reference. Standards documents are not intended for examination. Refer to section 7 - References for more information on Standards.

0.9 Level of Detail

The level of detail in this syllabus allows internationally consistent courses and exams. To achieve this goal, the syllabus consists of:

- General instructional objectives describing the intention of the Digital Accessibility Testing, Specialist Level
- A list of terms that students must be able to recall
- Learning objectives for each knowledge area, describing the cognitive learning outcome to be achieved
- A description of the key concepts, including references to sources such as accepted literature or standards

The syllabus content does not describe the entire knowledge area of software digital accessibility testing; it reflects the level of detail to be covered in Digital Accessibility Testing, Specialist Level training courses. It focuses on test approaches and test techniques that can apply to all software projects, including those

© International Software Testing Qualifications Board

following Agile software development.

The syllabus uses the terminology (i.e. the name and meaning) of the terms used in software testing and quality assurance according to the ISTQB® Glossary (ISTQB® Glossary, Chapter 7.3)

For the terminology in related disciplines please refer to the respective glossaries: IREB-CPRE for requirements engineering (IREB-CPRE, Chapter 7.3), and Agile Alliance for agile terminology (Agile Alliance, Chapter 7.3)

0.10 How this Syllabus is Organized

There are six chapters with examinable content. The top-level heading for each chapter specifies the time for the chapter; timing is not provided below the chapter level. For accredited training courses, the syllabus requires a minimum of 675 minutes (11.25 hours) of instruction, distributed across the six chapters as follows:

- Chapter 1: 90 minutes – Basic Concepts
  - The tester learns the basic concepts related to digital accessibility
  - The tester understands the different categories of disabilities
  - The tester understands the legal risks and impact of not applying digital accessibility to software
- Chapter 2: 75 minutes – Digital Accessibility Standards
  - The tester learns digital accessibility guidelines and their applicability
  - The tester understands digital accessibility regulations
- Chapter 3: 120 minutes – Digital Accessibility Testing Activities
  - The tester understands the approach to digital accessibility testing
  - The tester learns the digital accessibility test activities
  - The tester learns the typical project risks and common digital accessibility defects
- Chapter 4: 210 minutes – Digital Accessibility Testing lifecycle
  - The tester learns about the digital accessibility test preparation
  - The tester can identify the applicable digital accessibility testing tasks
  - The tester can analyze the digital accessibility testing results
  - The tester can apply best practices to report and communicate test results
- Chapter 5: 75 minutes – Assistive Technology
  - The tester understands assistive technologies
  - The tester can utilize assistive technologies in digital accessibility testing

© International Software Testing Qualifications Board

- Chapter 6: 105 minutes – Tools for Digital Accessibility Testing
  - The tester understands the types of digital accessibility testing tools
  - The tester understands the categories of digital accessibility testing tools available
  - The tester understands how to integrate tools in the software development lifecycle for digital accessibility testing

© International Software Testing Qualifications Board

1. Basic Concepts – 90 minutes

Keywords

accessibility, digital accessibility, accessibility evaluation, interaction capability, usability, user experience

Domain-specific keywords

medical model of disability, social model of disability

Learning Objectives for Chapter 1:

1.1 Fundamentals of Digital Accessibility

DAT-1.1.1 (K2) Explain the concept of digital accessibility

DAT-1.1.2 (K2) Explain the objectives of a digital accessibility evaluation

DAT-1.1.3 (K2) Summarize key principles and practices for digital accessibility testing

1.2 Disabilities

DAT-1.2.1 (K2) Explain the different categories of disabilities

DAT-1.2.2 (K2) Explain the importance of accessible software

1.3 Risks of Inadequate Digital Accessibility

DAT-1.3.1 (K2) Summarize the risks and impact of not applying digital accessibility to software

1.1 Fundamentals of Digital Accessibility

This section considers the concepts of digital accessibility and its relationship with interaction capability (the updated ISO/IEC 25010:2023 quality characteristic traditionally referred to as usability) and user experience.

1.1.1 Digital Accessibility

Digital accessibility concerns whether people with disabilities can perceive, understand, navigate, interact with, and use digital products and services. Accessibility can be affected by barriers that make use difficult as well as barriers that prevent use altogether.

Conformance to defined accessibility standards and guidelines, such as the Web Content Accessibility Guidelines [W3C WCAG], provides an objective basis for evaluating specified accessibility requirements. Conformance does not, by itself, demonstrate that a product is free of accessibility barriers for every user or context of use. WCAG is organized around four principles: perceivable, operable, understandable, and robust (POUR).

Digital accessibility overlaps with usability (referred to as interaction capability in ISO/IEC 25010:2023) and user experience, but the concepts are not interchangeable. Digital accessibility focuses on barriers

© International Software Testing Qualifications Board

that affect people with disabilities and on applicable accessibility requirements; usability addresses the effectiveness, efficiency, and satisfaction of specified users in a specified context of use; and user experience addresses users' perceptions and responses resulting from use or anticipated use. A product may conform to specified accessibility requirements and still present usability or user-experience barriers. For a comprehensive treatment of usability evaluation, refer to the ISTQB Specialist Syllabus, Usability Testing [ISTQB_CT_UT].

Examples of digital accessibility defects:

- A website uses text in an image to convey information, without alternative text available or with alternative text that does not match the information in the image. A screen reader user, therefore, does not have access to the same information as other users.
- A software product does not support navigation via keyboard, only via mouse. Users unable to use a mouse can therefore not interact with the software product.
- An online banking application has a strict, non-adjustable 60-second timeout for security. A user with limited mobility or hand tremors cannot complete the transaction within this limited timeframe, effectively locking them out of their account.

1.1.2 The Objectives of a Digital Accessibility Evaluation

A digital accessibility evaluation can combine digital accessibility testing and end-user testing. End-user testing is discussed further in Sections 3.2.4 and 3.3.

A digital accessibility evaluation examines whether people with disabilities encounter barriers when perceiving, navigating, interacting with, or using the software product, and whether applicable accessibility requirements are satisfied.

The key objectives of a digital accessibility evaluation, usability (interaction capability) evaluation, and user experience evaluation are compared in the following table. Usability evaluation is addressed in more detail in the ISTQB Specialist Syllabus, Usability Testing [ISTQB_CT_UT].

© International Software Testing Qualifications Board

Table 1: Objectives of Digital Accessibility, Interaction Capability, and User Experience Evaluations

Type of evaluation	Target group	Key objective
Digital accessibility evaluation	Users with disabilities	Evaluate the direct interaction between users and the software product, focusing on understanding problems related to digital accessibility barriers, rather than general efficiency or satisfaction.
Interaction capability evaluation	All users	Evaluate the direct interaction more broadly between users and the software product.
User experience evaluation	All users	Evaluate the services received prior to, during, and after the use of the software product to measure overall satisfaction and emotional response.

The principal techniques applied in these evaluations are shown in the following table:

Table 2: Principal Evaluation Techniques

Technique	Users involved?	Key characteristic	Evaluation Type
Digital accessibility testing	Optionally	Experience experts are observed while they perform typical tasks with the software product, or they perform digital accessibility evaluation independently and provide feedback.	Qualitative, Quantitative
Usability review	Optionally	Experts and users evaluate the user interface of a software product for interaction capability problems; the evaluation is based on their experience.	Qualitative
Usability testing	Yes	Users are observed while they perform typical tasks with the software product.	Qualitative, Quantitative
User surveys	Yes	Users fill out questionnaires regarding their satisfaction with the software product.	Qualitative, Quantitative

1.1.3 Principles and Practices for Digital Accessibility Testing

The following principles and practices support effective digital accessibility testing:

1. Define the target conformance level for digital accessibility. Establish the target based on applicable legal, regulatory, contractual, organizational, or project requirements. WCAG 2 defines three conformance levels: A, AA, and AAA. Level AA is a common target in many policies and regulations, but the required WCAG version and conformance level depend on the applicable context (see section 2.2.3).

© International Software Testing Qualifications Board

2. Create or adapt guidelines for accessible design

Guidelines must comply with applicable legal requirements and align with the chosen target conformance level. It is recommended to establish a dedicated support channel or network of digital accessibility Subject Matter Experts (SMEs) where digital accessibility questions from the development teams can be answered competently within an agreed timeframe.

3. Educate development and management teams and implement preventative measures

Implement preventative measures so that development teams can avoid building software with digital accessibility defects. This is particularly important for applicable legal requirements. Training should include factors such as:

- Legal requirements for digital accessibility
- Guidelines for accessible design and how to interpret and apply them
- Tools and techniques to use when evaluating digital accessibility
- The relationship between interaction capability and digital accessibility

4. Digital accessibility testing focuses on the following aspects:

- Focus on understanding the tools, techniques, and practices
- Place emphasis on identifying and understanding digital accessibility barriers and their causes, rather than treating efficiency or satisfaction as the primary accessibility criterion
- Use tasks that concentrate on specific areas of concern for potential digital accessibility defects, and follow the digital accessibility plan

A digital accessibility evaluation is similar in nature to usability evaluation but focuses specifically on accessibility barriers affecting users with disabilities and on applicable digital accessibility requirements. A digital accessibility evaluation should consider relevant digital accessibility standards, which are listed in Chapter 2.

1.2 Disabilities

1.2.1 Categories of Disabilities

For the purposes of digital accessibility testing, this syllabus focuses on five functional disability categories:

1. Visual Disabilities
2. Hearing Disabilities
3. Mobility/Physical Disabilities
4. Cognitive and Learning Disabilities
5. Speech Disabilities

© International Software Testing Qualifications Board

1. Visual Disabilities

Visual disabilities are sensory disabilities with a range of variations based on visual acuity and field of vision. Vision loss, or blindness, is considered a disability if vision cannot be fully corrected with standard glasses, contact lenses, medication, or surgery.

Types of blindness can range between total loss of vision (very rare) to seeing shapes. Visual disabilities also include color blindness, or color deficiency, in which colors are perceived in a different way.

Examples of barriers:

- Software that does not support keyboard use
- Incorrect usage of semantic mark-up for integration with assistive technology such as screen readers.
- Platforms reliant solely on visuals without text alternatives
- Unresponsive software that doesn't support zoom or enlargement of text without loss of content
- System states that rely exclusively on color to convey the meaning (e.g., a form where incorrect fields are only indicated by a red border without an accompanying error icon or text message).

2. Hearing Disabilities

Hearing disabilities comprise a form of hearing loss, in which it can be difficult or impossible to hear sounds such as speech. Hearing ability can range from partially impaired hearing to full hearing loss in one or both ears. Auditory processing disorder (APD) is a neurodevelopmental disorder that causes difficulty to process sounds such as speech.

Examples of barriers:

- Videos without closed captions
- Audio without a written transcript
- Transcripts that do not match what is said in the recorded fragment, causing a disbalance in shared information

3. Mobility/Physical Disabilities

A mobility disability refers to a physical condition that limits a person's ability to move. Mobility impairment includes limb loss or disability, reduced manual dexterity such as fine motor control, difficulties with coordination between body systems such as organs, or issues affecting the skeletal structure. It may result from a genetic condition, an illness or injury, or the natural effects of aging.

Examples of barriers:

- Software that doesn't support interaction with software products via keyboard, mouse, or assistive technology (AT) such as a mouth stick.
- Difficulty clicking small buttons due to tremors or use of assistive technology such as a customized mouse or mouth stick

4. Cognitive and Learning Disabilities

Cognitive disabilities concern the brain and can be caused by traumatic injuries or can occur by themselves. Cognitive disabilities concern a wide range of disabilities and disorders, with each

© International Software Testing Qualifications Board

experience unique to the affected person. This makes designing for cognitive disabilities particularly difficult, as what helps one person with a certain diagnosis may not help another with the same diagnosis.

Examples of barriers:

- Sole use of written language without supporting visuals such as icons
- Sole indication via symbols without supporting text
- Use of complex language, such as colloquial or idiomatic language, without supporting visuals or plain text
- Time limits
- Use of flashing or animations that cannot be turned off
- Complex layout and navigation

5. Speech Disabilities

Speech disabilities can include stuttering, dysarthria or slurred speech, and conditions that result in a partial or complete inability to speak. They may result from neurological or physical conditions or may occur with other disabilities.

Examples of barriers:

- A software product relies solely on voice activation
- An IVR (Interactive Voice Response) with voice as sole form of input

Disability characteristics can also vary by duration and context. A functional disability may be permanent, temporary, or progressive, while situational conditions can create similar barriers in a particular environment. For example, a broken arm may temporarily affect mobility, bright sunlight may create a situational visual barrier, and progressive changes in vision or motor control may alter accessibility needs over time. These characteristics can affect how users interact with software but are not separate functional disability categories.

1.2.2 The Importance of Accessible Software

Disabilities and accessibility barriers may also vary by duration and context. A limitation may be permanent or temporary; a barrier may be situational; and some disabilities are progressive, so accessibility needs can change over time. Two disability models are particularly relevant to digital accessibility:

1. Medical model of disability
2. Social model of disability

Accessible software enables people with disabilities to obtain information, complete tasks, and participate in digital services with fewer barriers. It can also support legal and contractual obligations, reduce remediation effort, and broaden access to products and services. Disability models provide useful

© International Software Testing Qualifications Board

perspectives for understanding why barriers arise and where responsibility for removing them lies. The two models emphasized in this syllabus are:

1. Medical model of disability

This model views disability as an issue that exists within the person, caused by injury, illness, or health condition. The impairment is seen as the root cause of the problem, and the aim is to diagnose, treat, manage, or “fix” it. Under this model, solutions focus on the individual adapting to the world, rather than the world adapting to them. In this context, assistive technology is often viewed strictly as a medical tool or personal aid meant to compensate for the individual's “deficit”, placing the burden on the user to find and utilize personal adaptive strategies to interact with standard, unmodified software.

2. Social model of disability

Popular in the disabled community, this model shifts focus away from the impairment and places responsibility on society. It argues that people are disabled not by their conditions, but by the physical, attitudinal, and systemic barriers that prevent them from participating equally. For example, a wheelchair user is not inherently limited by their condition, but by buildings without ramps or elevators. The goal here is to redesign environments, laws, and digital interfaces so that everyone can participate fully. Digital accessibility is strongly aligned with the social model of disability. In software development, this means building interfaces that proactively support a wide range of assistive technologies (e.g., screen readers or switch devices) and enable users’ preferred adaptive strategies (such as zooming in, changing color contrast, or relying on keyboard navigation). The software adapts to the user, not the other way around.

1.3 Risks of Inadequate Digital Accessibility

1.3.1 Risks and Impact of Not Applying Digital Accessibility to Software

Digital accessibility testing acts as a critical mitigation strategy to identify and resolve the sources of risk before they manifest into negative business impacts.

The risks associated with digital accessibility can be divided into two main categories: the sources of the risk (why the software fails) and the risk impacts (the consequences of releasing inaccessible software).

Sources of digital accessibility risk (project and technical risks)

The following root causes could introduce digital accessibility barriers into the software development lifecycle (SDLC). By mitigating these sources through testing, teams prevent larger impacts downstream:

1. Organizational risks
- Lack of qualified specialists in digital accessibility within the organization
- Insufficient knowledge of basic digital accessibility principles among designers, developers, and testers
- Insufficient management attention to digital accessibility test results, often incorrectly dismissing them as edge cases
- Inappropriate release decisions by management, such as refusing to postpone a release even when critical digital accessibility blockers are found
- Poor, vague, or entirely missing non-functional requirements regarding digital accessibility
- Failing to allocate time and budget for digital accessibility in the project test plan

© International Software Testing Qualifications Board

2. Technical & Maintenance risks
   - Using non-standard solutions (e.g., building custom UI components instead of using semantic, native HTML elements) that assistive technologies cannot interpret.
   - Inaccessible design patterns that make future digital accessibility fixes complex and costly
   - Software architecture that does not support digital accessibility principles
   - Compatibility issues with assistive technologies
   - Inconsistent digital accessibility support across platforms, devices, and browsers, leading to uneven user experience
   - No digital accessibility evaluation performed, in particular no digital accessibility test

3. Supplier risks
   - Third-party component suppliers or external vendors do not have the required qualifications in digital accessibility development or testing
   - Suppliers do not follow agreed-upon digital accessibility guidelines (e.g., WCAG) when delivering code
   - Digital accessibility testing results provided by external suppliers are inaccurate, delivered late, or missing entirely

Risk impacts of inaccessible software (product and business risks)

The following negative consequences could occur when digital accessibility risks are not mitigated, and inaccessible software reaches the end-user:

1. Legal & Regulatory risks
   - Non-conformance with applicable digital accessibility laws, regulations, standards, or contractual requirements
   - Lawsuits, fines, or regulatory penalties in jurisdictions with strong digital accessibility requirements
   - Increased legal exposure if competitors or users file formal complaints

2. Interaction Capability & User Experience risks
   - Complete exclusion of entire user groups from critical digital services
   - Increased frustration for users relying on assistive technologies
   - Higher abandonment rates if people cannot complete core tasks

3. Brand & Reputation risks
   - Perception of the product/company as non-inclusive, discriminatory, or out of touch
   - Negative publicity from digital accessibility advocacy groups, media, or social media campaigns
   - Loss of customer trust and loyalty not only from users with disabilities, but also from allies and diversity-conscious consumers

4. Market & Financial risks
   - Missed market share - excluding users with disabilities means losing access to a large, often overlooked customer base
   - Increased retrofit costs if digital accessibility is added late in development instead of early
   - Lost contracts with organizations that require digital accessibility compliance from vendors

© International Software Testing Qualifications Board

2. Digital Accessibility Standards – 75 minutes

Keywords

(none)

Domain-specific keywords

Web Content Accessibility Guidelines

Learning Objectives for Chapter 2:

2.1 Digital Accessibility Guidelines

DAT-2.1.1 (K2) Explain the general content and applicability of global guidelines to particular types of projects/applications

2.2 Digital Accessibility Regulations and Standards

DAT-2.2.1 (K2) Explain standards and legislation

DAT-2.2.2 (K2) Explain the general content and applicability of the ISO 9241-171 Standard

DAT-2.2.3 (K2) Explain the structure of the Web Content Accessibility Guidelines and other applicable guidelines for software digital accessibility

DAT-2.2.4 (K2) Explain typical criteria described in the Web Content Accessibility Guidelines

Please note that only the information explicitly provided in this chapter is examinable and not the entire contents of the standards themselves.

2.1 Digital Accessibility Guidelines

Digital accessibility guidance can take several forms. Guidelines provide principles, recommendations, and/or testable criteria intended to improve accessibility. Standards formalize requirements or guidance through a recognized standards body, while legislation and regulation can make particular requirements legally applicable within a defined jurisdiction and scope. Whether a guideline or standard is mandatory depends on the applicable legal, regulatory, contractual, procurement, or organizational context.

Applicable legislation, regulation, procurement rules, contracts, or organizational policy may identify which accessibility standards or guidelines apply and which conformance target is required. Testable criteria can support defect detection, evaluation, auditing, and reporting. Guidance documents may also provide examples and techniques for meeting accessibility requirements.

Examples of universal digital accessibility guidelines:

- For mobile applications, provide large enough touch targets

© International Software Testing Qualifications Board

- Use short sentences and familiar words (plain language)
- Provide color-blind friendly design (i.e., do not rely only on color to convey information)

Many legal and policy frameworks reference WCAG or standards that incorporate WCAG. The applicable WCAG version and conformance level depend on the jurisdiction, policy, contract, or other governing requirement [W3C WCAG].

2.2 Digital Accessibility Regulations and Standards

2.2.1 Standards and Legislation

A digital accessibility standard establishes defined accessibility requirements and/or guidance for products, services, content, or information and communication technology (ICT). Standards can be referenced by legislation, regulation, procurement requirements, contracts, or organizational policies.

Accessibility legislation and regulations vary by jurisdiction and scope. The following examples illustrate how standards, guidelines, and legal requirements can interact; they are not a complete statement of legal obligations.

European Union: Web Accessibility Directive, EN 301 549, and European Accessibility Act

The European Union uses legislation together with technical standards to establish accessibility obligations for particular public-sector bodies, products, and services.

Web Accessibility Directive (WAD) and EN 301 549

Directive (EU) 2016/2102, commonly called the Web Accessibility Directive (WAD), addresses the accessibility of websites and mobile applications of public sector bodies. The harmonized European standard EN 301 549 V3.2.1 (2021-03), Accessibility requirements for ICT products and services, provides technical accessibility requirements used in support of the WAD. EN 301 549 applies more broadly to ICT and contains requirements beyond WCAG.

For websites, EN 301 549 V3.2.1 draws heavily from WCAG 2.1 and also includes additional requirements relevant to the WAD. The applicable standard and version should be checked against the current official EU and ETSI sources because harmonized standards can be updated.

European Accessibility Act (EAA)

Directive (EU) 2019/882, commonly called the European Accessibility Act (EAA), establishes accessibility requirements for specified products and services placed on or provided in the EU market. Its scope includes, among other things:

- computers and operating systems
- ATMs, ticketing and check-in machines
- smartphones
- TV equipment related to digital television services
- telephony services and related equipment

© International Software Testing Qualifications Board

- access to audio-visual media services such as television broadcast and related consumer equipment
- services related to air, bus, rail and waterborne passenger transport
- banking services
- e-books
- e-commerce

EU Member States were required to transpose the EAA into national law by 28 June 2022. Its accessibility requirements generally apply from 28 June 2025, subject to the Directive's scope, exceptions, and transitional provisions.

United Kingdom: Equality Act and Public Sector Accessibility Regulations

The Equality Act 2010 protects disabled people from discrimination and includes duties relating to reasonable adjustments for services and public functions. The more specific public-sector web and mobile accessibility regulations build on these broader obligations.

The Public Sector Bodies (Websites and Mobile Applications) (No. 2) Accessibility Regulations 2018 apply to covered UK public-sector websites and mobile applications and require an accessibility statement, subject to specified exemptions and exceptions.

Current UK government guidance states that covered public-sector websites and mobile applications should meet WCAG 2.2 Level AA and publish an accessibility statement explaining their accessibility status. The current official guidance should be consulted because monitoring practice and referenced standards can change.

The Equality Act and the public-sector accessibility regulations have different scopes. Testers should therefore identify which legal or contractual requirements apply to the specific organization, product, and service being evaluated.

United States: Americans with Disabilities Act and Section 508

The Americans with Disabilities Act (ADA) prohibits disability discrimination. Title II applies to state and local governments, and Title III applies to businesses open to the public. The U.S. Department of Justice has consistently stated that ADA accessibility obligations extend to web content. A 2024 Title II rule establishes WCAG 2.1 Level AA as the technical standard for covered state and local government web content and mobile applications. Title III does not currently prescribe one detailed federal web technical standard, although businesses open to the public must still meet the ADA's nondiscrimination and effective-communication obligations.

Section 508 of the Rehabilitation Act applies to U.S. federal agencies when they develop, procure, maintain, or use information and communication technology. The Revised 508 Standards incorporate WCAG 2.0 Level A and AA success criteria for covered web and non-web electronic content and include additional requirements for software, hardware, authoring tools, and functional performance.

2.2.2 General content and applicability of ISO 9241-171:2025

ISO 9241-171:2025 – Ergonomics of human-system interaction — Part 171: Software accessibility

© International Software Testing Qualifications Board

ISO 9241-171:2025 applies to the accessibility of interactive systems and addresses a broad range of software, including web, mobile, office, home, learning-support, and other interactive systems.

The standard specifies requirements and gives guidelines for designing accessible software for people with a wide range of physical, sensory, and cognitive capabilities. It can be used as a source of accessibility requirements and design guidance when planning and evaluating software.

- The 2025 edition replaces ISO 9241-171:2008. Testers and training providers should use the current edition when consulting the standard and should verify whether later amendments or revisions have been published.
- Because the full ISO standard is referenced rather than reproduced here, only the concepts explicitly summarized in this syllabus are examinable.

2.2.3 The Web Content Accessibility Guidelines

The Web Content Accessibility Guidelines (WCAG) are W3C standards developed through the Web Accessibility Initiative (WAI). WCAG 2.2 is the current WCAG 2 Recommendation and W3C encourages use of the latest WCAG version for new accessibility work. WCAG 2.0 and 2.1 remain existing standards and may still be the versions referenced by particular laws, regulations, policies, or contracts [W3C WCAG].

WCAG 2 is organized in layers of guidance:

- Principles: four foundational principles — perceivable, operable, understandable, and robust (POUR).
- Guidelines: 13 high-level goals under the four principles in WCAG 2.2. The guidelines provide a framework but are not themselves the testable conformance requirements.
- Success criteria: testable statements associated with the guidelines. Success criteria are assigned Level A, AA, or AAA.
- Techniques and supporting guidance: informative resources describing ways to meet or understand success criteria; these are not themselves normative WCAG requirements.

WCAG conformance is determined by satisfying the applicable success criteria and the WCAG conformance requirements. WCAG itself does not create legal obligations unless it is incorporated into an applicable law, regulation, policy, contract, or other requirement.

The following tables summarize the three WCAG 2 conformance levels and give illustrative WCAG 2.2 examples:

Level A

© International Software Testing Qualifications Board

Table 3: WCAG Level A Conformance Examples

Description	Examples
The lowest WCAG conformance level. To conform at Level A, all applicable Level A success criteria must be satisfied.	Example: Success Criterion 1.1.1 Non-text Content requires text alternatives for non-text content, subject to stated exceptions.

Level AA

Table 4: WCAG Level AA Conformance Example

Description	Example
To conform at Level AA, all applicable Level A and Level AA success criteria must be satisfied. Level AA is referenced by many accessibility policies and regulations, but the required target depends on the governing context.	Example: Success Criterion 1.4.3 Contrast (Minimum) establishes minimum contrast requirements for text and images of text, subject to stated exceptions.

Level AAA

Table 5: WCAG Level AAA Conformance Example

Description	Example
The highest WCAG conformance level. To conform at Level AAA, all applicable Level A, Level AA, and Level AAA success criteria must be satisfied. WCAG notes that even AAA conformance does not address every user need.	Example: Success Criterion 1.4.6 Contrast (Enhanced) establishes enhanced contrast requirements for text and images of text, subject to stated exceptions.

Applying WCAG to non-web technologies

WCAG was written for web content. For non-web documents and software, W3C publishes Guidance on Applying WCAG 2 to Non-Web Information and Communications Technologies (WCAG2ICT). WCAG2ICT is informative guidance: it explains how WCAG 2.0, 2.1, and 2.2 Level A and AA success criteria can be interpreted for non-web ICT, but it does not itself set requirements. Applicable legislation or standards may impose additional or different requirements [W3C WCAG2ICT].

© International Software Testing Qualifications Board

Typical WCAG Success Criteria

Typical WCAG success criteria address barriers such as the following. The exact criterion and conformance level depend on the applicable WCAG version and target:

- Text alternatives for non-text content so equivalent information can be conveyed in another form.
- Keyboard accessibility so functionality can be operated without requiring a pointing device.
- Sufficient contrast and adaptable presentation so content can be perceived in different visual conditions.
- Meaningful structure, labels, instructions, and programmatic relationships so content and controls can be understood by users and assistive technologies.
- Error identification and assistance so users can recognize and recover from input errors.
- Compatibility information such as accessible names, roles, states, and values for user interface components.

© International Software Testing Qualifications Board

3. Digital Accessibility Testing Activities – 120 minutes

Keywords

digital accessibility audit, compliance, end-user testing, shift left, user experience

Domain-specific keywords

Accessible Rich Internet Applications (ARIA), CI/CD pipeline, document object model (DOM), linter

Learning Objectives for Chapter 3:

3.1 The Approach to Digital Accessibility Testing

DAT-3.1.1 (K2) Explain the planning activities for establishing a digital accessibility test

3.2 Digital Accessibility Testing Activities

DAT-3.2.1 (K2) Explain the benefits of shift left in digital accessibility testing

DAT-3.2.2 (K2) Explain how to perform digital accessibility testing

DAT-3.2.3 (K2) Summarize the process of digital accessibility audits as a method for evaluating conformance

DAT-3.2.4 (K2) Explain the relevance of end-user testing in evaluating real-world digital accessibility

3.3 Step-By-Step Approach to End-User Testing

DAT-3.3.1 (K2) Explain the purpose and approach adopted in end-user testing

3.4 Risks and Defects in Digital Accessibility Testing

DAT-3.4.1 (K2) Explain the typical project risks and mitigation measures in digital accessibility testing

DAT-3.4.2 (K2) Explain common software defects identified during digital accessibility testing

© International Software Testing Qualifications Board

3.1 The Approach to Digital Accessibility Testing

This chapter describes core activities used to evaluate digital accessibility. It covers planning, shift left, automated and manual testing, assistive-technology-supported testing, audits, and the involvement of people with disabilities.

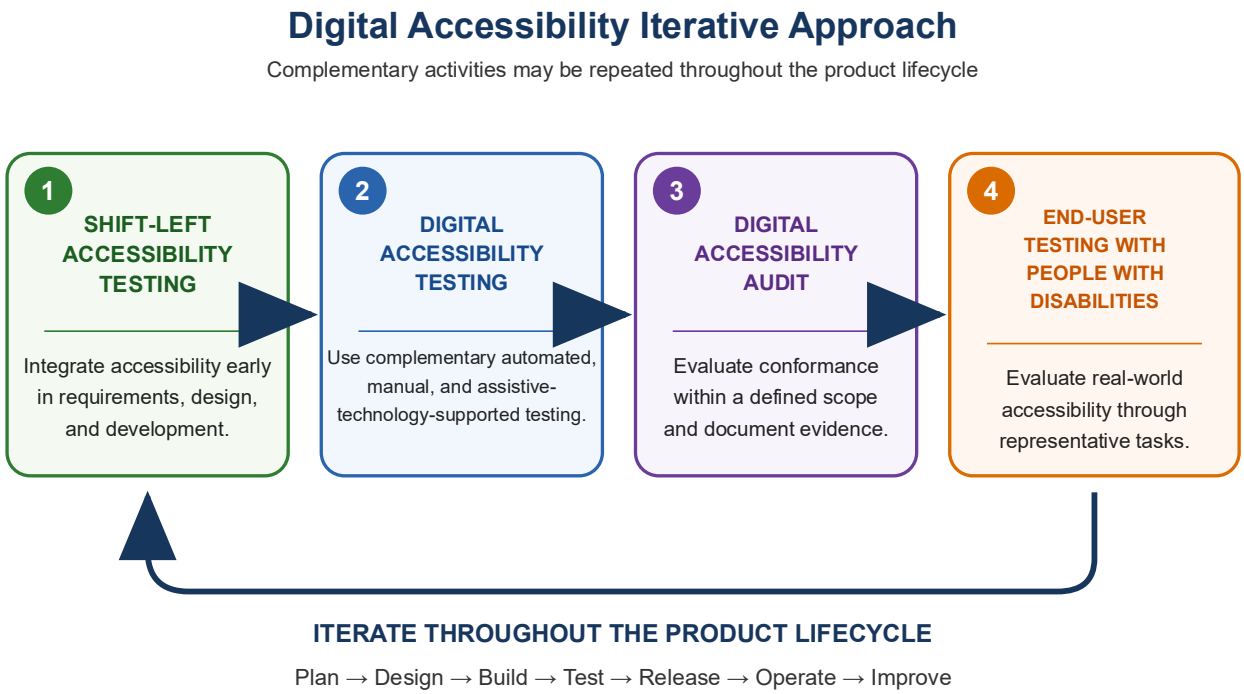


Figure 3-1: Digital Accessibility Iterative Approach

Figure 3-1 illustrates an iterative approach to digital accessibility testing. It combines early accessibility activities through shift left, ongoing digital accessibility testing, formal accessibility audits, and end-user testing with people with disabilities. These complementary activities provide different forms of evidence and may be repeated throughout the product lifecycle.

3.1.1 Planning Activities for Digital Accessibility Testing

Principal activities in digital accessibility test planning include:

- Prioritizing functionality and coverage
- Selecting tools
- Defining digital accessibility acceptance criteria

© International Software Testing Qualifications Board

- Building know-how

Functionality prioritization and coverage

Determining which areas of the application have the highest risk to users with disabilities. High-traffic areas (e.g., homepages), critical paths (e.g., checkout, login), and complex interactive components (e.g., carousels, modals) are to be prioritized.

Selecting Tools

Deciding on the set of tools. This involves selecting automated linting tools as part of the static code analysis for code-level tests, considering browser extensions for rapid page scanning and analysis, assistive technologies for screen reading, and other tools relative to the supported assistive peripherals.

Defining Acceptance Criteria for Digital Accessibility

Establish clear, measurable acceptance criteria based on the applicable accessibility requirements, risk, product context, and organizational quality policy. Completion criteria should define the required coverage and the treatment of unresolved findings; they should not be reduced to a single universal defect-count threshold.

Building Know-how for Digital Accessibility Testing

Identifying if external consultants are needed and at what phase of the project, or if the internal quality assurance team requires training on specific digital accessibility testing.

3.2 Digital Accessibility Testing Activities

This section describes the primary activities involved in testing for digital accessibility, from integrating testing early in the development lifecycle to performing formal audits and involving end-users to increase confidence in genuine digital accessibility.

3.2.1 Benefits of Shift Left in Digital Accessibility Testing

Shift left is the practice of moving test activities to earlier phases in the SDLC. Traditionally, testing, including digital accessibility testing, was often performed late in the SDLC, just before release. Shift left means integrating digital accessibility considerations and testing from the very beginning and continuously during requirements, design, and development phases.

The benefits of adopting a shift left for digital accessibility testing are significant:

- Reduced rework and remediation effort. Identifying accessibility issues during requirements, design, or early development can reduce the amount of rework needed later. Late discovery can require broader design, code, regression testing, or deployment changes. For example, correcting

© International Software Testing Qualifications Board

a color-contrast choice in a design system before implementation can avoid repeated changes across implemented screens.

- Improved product quality and innovation. When digital accessibility is considered from the start, this leads to a more robust, higher-quality, and consistently accessible user experience with a built-in digital accessibility solution. It encourages the team to think inclusively, which can spark innovation and result in solutions that benefit all users, not just those with disabilities.
- Enhanced team awareness and culture. Integrating digital accessibility into daily activities (e.g., design reviews, code reviews, test case reviews) fosters a culture where digital accessibility is a shared responsibility. It builds knowledge and empathy across the entire team, making digital accessibility assurance a standard part of the definition of ready and the definition of done.
- Faster time-to-market. By catching and resolving digital accessibility defects early and continuously, teams can avoid digital accessibility-related high-severity defects that can delay a product launch, allowing for more predictable and faster release schedules.
- Proactive risk mitigation. Addressing digital accessibility continuously reduces the legal risk associated with non-compliance with regulations such as the Americans with Disabilities Act (ADA) or the European Accessibility Act (EAA) (see chapter 2.2). It also mitigates the risk of reputational damage that can result from releasing an inaccessible product.

3.2.2 How to Perform Digital Accessibility Testing

Performing digital accessibility testing is not a single activity but a multi-faceted process that combines automated, manual, and assistive technology test activities. A comprehensive strategy utilizes all three to achieve broad coverage.

Automated testing

Automated testing uses software tools to evaluate source code or rendered interfaces against rules that can be checked programmatically. Such tools can efficiently identify some classes of potential accessibility non-conformance, but their coverage depends on the technology, ruleset, and context.

Automated tools can be integrated into test automation frameworks as third-party libraries, used as browser extensions for quick tests, built into the continuous integration/continuous deployment (CI/CD) pipeline to catch defects automatically with every code change, or deployed for site-wide scans.

Automated tools detect only the issues that can be evaluated reliably by their implemented rules and available context. Human judgment remains necessary for many questions involving meaning, logical order, task flow, interaction, and real-world use. Automated results therefore provide evidence, not proof that a product is accessible.

Manual testing

This involves a tester systematically evaluating for digital accessibility defects that automated tools cannot identify. It requires a deep understanding of digital accessibility guidelines and how users interact with digital content.

Key manual test activities include:

© International Software Testing Qualifications Board

- Keyboard-only navigation. Testing that all interactive elements (e.g., links, buttons, forms) can be reached and operated using only certain keyboard keys (e.g., Tab, Shift+Tab, Enter, and arrow keys) and that the focus order is logical.
- Content magnification. Zooming in the page to a required percentage (depending on requirements) to verify that content reflows correctly without requiring horizontal scrolling.
- Code inspection. Manually reviewing the HTML to verify semantic structure (e.g., correct use of headings, lists, landmarks, etc.) and proper ARIA implementation [W3C ARIA].
- Forms and error handling. Verifying that all form fields are properly labeled and confirming that error messages are clear, descriptive, and programmatically associated with the correct field.

Assistive technology testing

This is a specialized form of manual testing where the tester uses the same assistive technologies that people with disabilities rely on. The goal is to evaluate how the product behaves when used through representative assistive-technology configurations. This testing can identify functional barriers, but it does not substitute for feedback from people with disabilities about their experience.

This typically involves testing with the most common screen readers (e.g., commercial tools, OS built-in tools, open-source tools) to verify if the content is read out logically, if interactive elements are announced correctly, and if the user can navigate and complete tasks successfully. It can also include testing with other assistive technology like speech recognition software or screen magnifiers.

Artificial Intelligence (AI) analysis

AI-assisted analysis can support accessibility testing by identifying patterns or suggesting findings that rule-based tools may not detect

3.2.3 Digital Accessibility Audits as a Method for Evaluating Conformance

A digital accessibility audit is a formal and comprehensive evaluation of a digital product (e.g., a website or mobile application) to determine its level of conformance with a specific set of digital accessibility standards, most commonly the WCAG. Unlike ongoing testing, an audit is a structured, point-in-time assessment that results in a formal report.

The primary purpose of an audit is to provide a documented, scoped assessment of conformance against specified accessibility requirements and to support prioritization and remediation. The strength of any conclusion depends on the defined scope, sample, methods, and evidence.

The process of performing a digital accessibility audit typically involves the following:

- Scope definition. This is to clearly define what will be audited. This includes:
  - The specific website, application, solution, or platform.
  - The targeted digital accessibility standard and version (e.g., WCAG 2.2, or an earlier WCAG version were required by the applicable policy, regulation, contract, or standard).
  - The targeted conformance level (A, AA, or AAA).
- Representative sample of pages/screens. It is based on business requirements and decisions. It can be that all pages/screens are candidates for audit. In some cases, it is impractical to audit every

© International Software Testing Qualifications Board

single page/screen of a large solution. Instead, a representative sample is selected. This sample should include:
  - The homepage and other high-traffic pages/screens.
  - Pages/Screens representing key user flows (e.g., login, search, checkout process, etc.).
  - Pages/Screens that contain a wide variety of content and components (e.g., forms, tables, multimedia, complex widgets, etc.).
  - Pages/Screens containing legal information and policies.
- Conformance evaluation. The audit team performs a thorough evaluation of the selected sample using a combination of automated, manual, and assistive technology test activities (as described in Section 3.2.2). Every applicable success criterion from the target standard is assessed against the sample pages/screens.
- Logging findings and prioritization of defects. Every identified defect is meticulously documented. Each finding should include:
  - A clear description of the defect and its location.
  - The applicable accessibility requirement or WCAG success criterion(s), when the finding represents non-conformance.
  - The impact of the defect on users with different types of disabilities.
  - Recommendations for remediation.
  - Defects are often prioritized by severity (e.g., critical, serious, moderate) to help development teams address the most impactful defects first.
- Audit report generation. The final deliverable is a formal report that summarizes the findings. This report typically contains:
  - An executive summary stating the scope and the conformance conclusion supported by the evaluation.
  - The scope and methodology of the audit.
  - A detailed list of all identified defects and recommendations for fixing them.
  - A concluding statement describing the evaluated product, scope, limitations, and principal accessibility findings.

3.2.4 The Relevance of End-user Testing in Achieving a Highly Functional Product

End-user testing in digital accessibility is a specialized form of validation that relies on the experience of individuals with disabilities. It involves observing actual users with disabilities as they attempt to complete tasks on a product using their own assistive technologies. While a digital accessibility audit is critical for verifying technical conformance, end-user testing is essential for evaluating true user experience.

Automated tools, manual expert evaluation, and standards-based conformance testing each have limitations. A product can satisfy evaluated technical criteria and still present barriers that become apparent during real tasks. End-user testing complements these activities by providing evidence from people with disabilities using the product in representative contexts.

End-user testing can provide the following additional value:

- Reveal real-world barriers: End-user testing uncovers interaction capability defects that automated tools and expert-led manual testing can miss. It provides insight into the practical challenges users face, which are influenced by their specific disability, their chosen assistive technology, and their personal interaction patterns. For example, a data table might be technically compliant, but a

© International Software Testing Qualifications Board

screen reader user might find it impossible to understand the relationships between headers and cells during a real-world task.

- Provides authentic context and empathy: Watching a real user struggle with a product provides invaluable context and builds empathy within the development team. Such direct feedback is far more powerful than a list of defects and helps teams to understand the "why" behind digital accessibility guidelines.
- Determines that solutions are effective: When a team implements a solution for a digital accessibility defect, end-user testing can validate whether the solution is genuinely effective and intuitive for the intended users.

End-user testing is particularly valuable for critical user journeys, novel or complex interactions, major changes, and validation of remediated barriers. It can be performed iteratively during design and development and again on sufficiently stable implementations. It complements, rather than replaces, standards-based and expert testing.

3.3 A Step-By-Step Approach to End-User Testing

3.3.1 The Purpose and Approach Adopted in End-user Testing

The purpose of end-user testing is to validate the user experience (UX) with the people for whom the digital accessibility features are designed. It follows the principle "Nothing about us without us".

The approach considers the following topics:

- Recruitment. Recruit participants with a diverse range of disabilities (visual, auditory, motor, cognitive) and different levels of technical proficiency.
- Protocol Design. Create tasks that focus on real user goals (e.g., "Find a product and buy it") rather than specific digital accessibility tests. Allow users to use their own devices and setup.
- Facilitation:
    - Moderated: A moderator observes the user (remotely or in-person) and asks questions. This provides the deepest insights.
    - Unmoderated: Users record themselves performing tasks. This is faster but offers less context on why a user struggled.
- Live observations and demonstrations. Make it as easy as possible for developers, designers, and managers to observe these sessions. Schedule them at convenient times or host live demonstrations where the team observes a user navigating the product with their assistive technologies. Witnessing real barriers firsthand is the fastest way to dissolve skepticism and secure team buy-in.
- Analysis. Classify findings by the user barrier, applicable requirement, affected task, and likely source. A finding may involve code, design, content, configuration, or user experience, and these categories can overlap.
- Compensation. Always compensate participants fairly for their time and expertise.

Reporting. Summarize observed barriers, affected tasks, user impact, supporting evidence, and follow-up actions while protecting participant privacy.

- Privacy and data protection. Obtain informed consent, collect only necessary personal data, and follow applicable privacy and data-protection requirements. Define how recordings and notes will

© International Software Testing Qualifications Board

be stored, accessed, retained, and deleted; anonymize or pseudonymize participant information where appropriate and avoid disclosing disability or medical information unnecessarily.

3.4 Risks and Defects in Digital Accessibility Testing

Digital accessibility testing is affected by project risks, and testers should identify and communicate those risks while working within the project's established risk-management responsibilities. Common software defects should also be recognized and reported.

- Project risks (risks within the test process or strategy)
- Common digital accessibility defects (frequent technical and structural code defects)

3.4.1 Typical Project Risks and Mitigation Measures

The most frequent project risks in digital accessibility evaluation include:

Table 6: Typical Project Risks and Mitigation Measures

Risk Name	Triggering Event	Risk Impact	Suggested Mitigation
Over-reliance on test automation	Using test automation tools as the sole method of digital accessibility verification.	Relying only on automated findings can leave context-dependent, interaction, and user barriers undetected, even when automated scans report no issues.	Implement a layered test strategy combining automated scanning with manual testing and testing with assistive technologies.
“Works on my machine” syndrome / Single-environment testing	Standardizing testing on only one screen reader and browser combination (e.g., a specific mobile platform configuration) and assuming it works universally on others. Various operating system combinations might have a big difference in overall digital accessibility, as assistive technologies behave differently across platforms.	Severe digital accessibility barriers occur for users on different platforms due to inconsistent assistive technology behavior across operating systems.	Configure test environments to cover the most common combinations of operating systems, browsers, and assistive technologies.

© International Software Testing Qualifications Board

Reliance on "accessibility overlay" solutions	Installing third-party "digital accessibility overlay" widgets instead of remediating the underlying code.	Overlay solutions may not correct underlying accessibility barriers and can introduce compatibility or interaction problems in some configurations.	Prioritize remediation in the product, content, design, or configuration and verify the result against applicable requirements and supported assistive technologies.
Late-level testing	Treating digital accessibility as a final "checkbox" activity immediately prior to release.	Important design or architectural barriers may be discovered late, increasing remediation effort and creating release risk.	Apply shift left by integrating appropriate accessibility activities during requirements, design, implementation, and testing.
Allowing opinions to override facts	Treating digital accessibility as a subjective design choice (e.g., "The contrast looks fine to me" or "Nobody uses a keyboard here").	Valid digital accessibility defects are dismissed during triage because strong opinions dominate the discussion.	Use reproducible evidence and applicable accessibility requirements during triage, supplemented by user feedback where appropriate.

3.4.2 Common Software Defects in Digital Accessibility Testing

The most frequent technical defects identified during digital accessibility evaluation include:

Table 7: Common Digital Accessibility Software Defects

Common Defect	Description
Incorrect or unnecessary ARIA [W3C ARIA]	ARIA can create barriers when roles, states, properties, or required interaction behavior are implemented incorrectly. Prefer native semantic controls when they meet the need; when custom controls are required, implement and test the applicable ARIA pattern and keyboard behavior.
Ignoring focus management	When a modal closes or the user interface updates dynamically, keyboard focus is lost or reset to the beginning of the view, causing a loss of context and disrupting keyboard navigation.

© International Software Testing Qualifications Board

Dynamic content blind spots	In a single page application, content updates without a page reload, but status changes (e.g., “Item added to cart”) fail to be announced programmatically to screen readers.
Vague alternative (alt) text	Images have an *alt* attribute, but the text is not meaningful to the user, e.g., accepting “image123.jpg”, “picture”, or redundant text like “Image of logo” as valid.
Zoom and reflow failures	At high zoom levels (e.g., 200% or 400%), content overlaps, disappears, or requires horizontal scrolling to be read.
Reliance on visual cues	Visual cues (like displayed error messages or red borders) fail to be conveyed programmatically or via text for users who cannot perceive color or see the interface visually.

© International Software Testing Qualifications Board

4. Digital Accessibility Testing Lifecycle – 210 minutes

Keywords

test planning

Domain-specific keywords

accessibility conformance report (ACR), digital accessibility remediation, screen reader, voluntary product accessibility template (VPAT)

Learning Objectives for Chapter 4:

4.1 Preparing a Digital Accessibility Test

DAT-4.1.1 (K2) Explain factors to consider in digital accessibility test planning

DAT-4.1.2 (K2) Explain the configuration requirements for digital accessibility tests

4.2 Executing Digital Accessibility Tests

DAT-4.2.1 (K2) Explain the applicable digital accessibility test tasks

4.3 Analyzing and Communicating Findings

DAT-4.3.1 (K4) Analyze digital accessibility test results to identify patterns, root causes, user impact, and priorities for action

DAT-4.3.2 (K2) Explain typical criteria associated with findings

DAT-4.3.3 (K3) Apply good practices to report and communicate digital accessibility test results

DAT-4.3.4 (K2) Summarize the purpose and structure of a formal digital accessibility report

4.1 Preparing a Digital Accessibility Test

Digital accessibility testing should start as early as appropriate within the software development lifecycle (SDLC) and continue at relevant points alongside other test activities. This chapter covers project-specific planning, environment configuration, execution, analysis, and communication of findings.

4.1.1 Factors to Consider in Digital Accessibility Test Planning

Integrating digital accessibility into test planning from the very beginning is crucial for efficiency and effectiveness. Rather than treating digital accessibility as a final separate test level or phase, it should be a consideration throughout the SDLC. While Chapter 3 outlines the high-level testing strategy and techniques, this section focuses on creating the specific, actionable test plan for a given project or release.

© International Software Testing Qualifications Board

A number of factors need to be considered when planning digital accessibility testing. These include the following:

- Identifying digital accessibility requirements
- Defining the test scope and the test objectives
- Determining the test approach
- Planning resources and test environments
- Defining roles and responsibilities
- Test scheduling and integration

Identifying digital accessibility requirements

A digital accessibility requirement is a requirement regarding the digital accessibility of a component or system. It provides the basis for evaluating whether a digital product meets the needs of users with disabilities. Digital accessibility requirements can originate from several sources:

- Project requirements. Requirements stated in product documentation, acceptance criteria, user stories, design-system rules, or organizational policy (e.g., “As a keyboard-only user, I must be able to navigate the checkout form without getting stuck”).
- External requirements. Requirements imposed or referenced by applicable legislation, regulation, standards, procurement rules, or contracts (e.g., “The application must conform to WCAG 2.2 Level AA”).
- Derived technical requirements. Requirements derived from an applicable standard, platform accessibility model, design system, or supported assistive-technology interaction even when the implementation detail is not repeated in a user story. Derived requirements should be traceable to their source rather than treated as undocumented assumptions.

Defining the test scope and the test objectives

- Conformance goals. Clearly state the target digital accessibility standard (e.g., WCAG) and the desired conformance level (A, AA, AAA).
- Jurisdictional requirements. Identify any specific legal or regulatory requirements (e.g., Section 508 in the U.S., EN 301 549 in Europe).
- In-scope/Out-of-scope. Define which parts of the application, platform, or user journeys will be tested. This may involve prioritizing critical user paths, such as registration and core functionality.

Determining the test approach

- Automated testing: Integrate appropriate test automation tools into development and CI/CD workflows to identify rule-checkable issues early, while recognizing that automated results do not cover all accessibility requirements or user barriers.
- Manual testing: Plan for test activities that require human judgment, such as keyboard navigation, and logical content flow.
- Testing using assistive technology: Planning for testing with specific screen readers that are integrated into operating systems or external ones, screen magnifiers, or voice control software

© International Software Testing Qualifications Board

- End-user testing: Involve people with disabilities in test activities to gain real-world feedback on user experience and digital accessibility.

Planning resources and test environments

- Team skills: Assess the team's knowledge of digital accessibility and plan for training, if needed. Awareness of disabilities is important.
- Tools and software: Allocate budget and time for manual testing, test automation tools, assistive technologies, representative devices/environments, and any external accessibility expertise or participant involvement required by the test approach.
- Test environment: Verify the test environment is configured to support digital accessibility testing (e.g., it is accessible for assistive technologies, and test automation).

Defining roles and responsibilities

Identifying who will perform which parts of the testing and remediation process to facilitate comprehensive coverage. Examples include:

- Tester: To execute routine automated and manual digital accessibility tests, report defects, and verify fixes (i.e., this is the core role executing the test plan).
- Developer: To consult on technical feasibility and implement accessible code and structural fixes.
- UX/UI designer: To verify that visual and interactive elements (e.g., color contrast, focus states) meet digital accessibility standards prior to development.
- Digital accessibility expert/consultant: To act as a Subject Matter Expert (SME), resolve complex WCAG interpretations, guide the team, and conduct formal compliance audits.
- Product owner / business analyst: To define explicit digital accessibility requirements, user stories, and acceptance criteria.
- Participant with a disability / end user: To provide first-hand feedback on barriers encountered during representative tasks and, when applicable, the use of assistive technologies. This role is distinct from the accessibility specialist or consultant role.

Scheduling and integration

- Entry and exit criteria: Define what must be ready before planned test activities begin and what evidence is required for completion. Exit criteria can include required coverage completed, agreed high-risk findings addressed or formally accepted, retesting completed, and required reports produced. Avoid relying on a universal percentage of defects resolved.
- Milestones: Integrate digital accessibility testing activities into project sprints and project milestones, from early design reviews through pre-production phases and testing in production.

4.1.2 Configuration Requirements for Digital Accessibility Tests

A well-configured test environment is the foundation of reliable and repeatable digital accessibility testing. The setup must reflect the technology used by the target audience, as support for assistive technology can vary significantly across platforms.

A typical test environment configuration includes:

© International Software Testing Qualifications Board

Table 8: Typical Digital Accessibility Test Environment Configuration

Operating systems (OS) and Browsers	Identify and prepare the most common combinations of operating system and browser for the user base.
Assistive technologies	Screen readers: Prepare the screen readers and versions relevant to the supported platforms and target users. Include representative configurations; default settings can be useful as a baseline but should not be assumed to represent every user. Screen magnifiers: Prepare built-in or supported magnification tools relevant to the target environment. Voice control software: Configure supported speech/voice-control tools for hands-free interaction where relevant. Other assistive technology: Include switch access, braille displays, alternative input, or other technologies when required by the target users, risks, or product context.
Browser extensions	Install trusted digital accessibility evaluation extensions to support rapid inspection during development and testing.
Device requirements	Testing on actual smartphones and tablets is crucial. Emulators cannot fully replicate the interaction between an operating system and its native assistive technology.
Test data	Make sure user accounts and data sets cover all relevant application states and user roles (e.g., guest versus authenticated user, administration versus standard user)

4.2 Executing Digital Accessibility Tests

Test execution combines test automation with manual evaluation, testing using assistive technologies, and, where planned, end-user testing. Each activity contributes different evidence and should be selected according to risk, scope, and the applicable requirements.

Digital accessibility test execution is not a single event. Activities can occur at different stages of development and need not follow one fixed sequence. Shift left places suitable checks earlier, while human-led evaluation and end-user testing are scheduled when representative functionality and environments are available.

© International Software Testing Qualifications Board

4.2.1 Identifying Applicable Digital Accessibility Test Tasks

The execution tasks below map to the automated, manual, assistive-technology, and end-user activities introduced earlier. They may be used at different points in the lifecycle rather than as a mandatory sequence:

1. Build-time automated testing
2. UI-based automated scanning
3. Manual digital accessibility testing

End-user testing, when included in the test approach

Build-time automated testing

Build-time automated checks can run when code is committed or a build is created (e.g., within a CI/CD pipeline). Static analysis tools such as linters can identify selected rule violations in source code and provide rapid feedback to developers before deployment.

UI-based automated scanning

Once the code is deployed to a development or test environment and a user interface (UI) can be present, a second layer of automated testing is performed. This involves running various helpers and tools or other test automation frameworks against the deployed solution. These tools analyze the rendered document object model (DOM) to find defects that are only apparent once the UI is built. The scan provides preliminary evidence about rule-checkable issues in the rendered interface. Manual and assistive-technology-supported testing can proceed according to risk and test readiness; they do not need to wait for every automated finding to be resolved.

Manual digital accessibility testing

This is a planned test activity that requires human judgment to uncover the major digital accessibility defects that specialized software cannot detect. On a stable build, a tester may systematically perform the following tests:

- Keyboard-only navigation testing
  Avoiding use of the pointing device (or disconnecting/disabling it where practical) to determine if there is a visible focus indicator, a logical focus order, no keyboard traps, and that all interactive elements (buttons, links, form fields, menus) can be operated using only standard keystrokes (e.g., Enter, Spacebar, Arrow keys).
- Screen reader testing
  Using a screen reader to evaluate how the product exposes structure, controls, states, and dynamic updates through the accessibility interface. This verifies that semantic HTML (headings, lists, landmarks) is used correctly, dynamic content updates (alerts, notifications) are announced without interrupting the user, and that images, form labels, and controls are announced accurately.
- Visual and content testing
  - Color contrast. Using analyzers to verify required contrast ratios based on the target conformance level (e.g., typically 4.5:1 for normal text at Level AA, and stricter ratios like 7:1 for Level AAA).
  - Zoom and reflow. Test text resizing and viewport reflow against the applicable accessibility requirements. For WCAG, verify the relevant success criteria and their

© International Software Testing Qualifications Board

stated exceptions rather than assuming every interface must always reflow into a single column without horizontal scrolling.

- Multimedia tests
  Verify applicable multimedia requirements such as captions, transcripts, and audio description.

  Safety-related visual content
  Evaluate flashing or other content that can cause physical reactions against the applicable accessibility criteria, including their conditions and exceptions.

4.3 Analyzing and Communicating Findings

Analysis and communication turn test findings into information that supports remediation and decision-making. Testers should consolidate evidence, analyze patterns and user impact, relate findings to applicable requirements, and communicate conclusions in forms appropriate to the audience.

4.3.1 Analyzing the Digital Accessibility Test Results

Digital accessibility analysis involves evaluating findings beyond simple defect listing. The analysis should identify patterns, likely root causes, user impact, recurring components or processes, and priorities for action within the defined evaluation scope.

Key analysis activities include:

Consolidating and verifying results	Gather raw findings from all testing activities (automated, manual, assistive technology, user feedback) into a single location. The tester must critically evaluate automated test results to filter out false-positive results and verify manual findings are reproducible.
Navigating and mapping to WCAG	Testers should be proficient in reading and applying the WCAG version relevant to the evaluation. WCAG structure: Principles (POUR) → Guidelines → Success Criteria (SC). W3C also publishes informative Understanding documents, techniques, and documented failures. Mapping findings: When a finding represents WCAG non-conformance, map it to the applicable success criterion or criteria. A finding can relate to more than one criterion, and some accessibility findings may relate to other standards, requirements, or user barriers rather than a WCAG violation. Legal status: WCAG is a technical standard. It has legal or contractual force only where incorporated by an applicable law, regulation, policy, contract, or other requirement. Supporting resources: Use W3C Understanding WCAG and How to Meet

© International Software Testing Qualifications Board

	WCAG resources to understand intent and implementation options. These supporting resources are informative rather than normative requirements.
Formulating actionable developer guidance	Explain the user barrier and the evidence, identify applicable requirements where relevant, and provide remediation guidance at the level appropriate to the tester's role. Guidance can include references to verified resources such as WAI-ARIA, the ARIA Authoring Practices Guide (APG), and relevant WCAG techniques. The tester need not prescribe a specific code implementation when design or development analysis is required.
Root cause analysis	Identify recurring or systemic patterns rather than viewing findings only in isolation. Consider whether repeated findings arise from a shared component, design pattern, content process, configuration, or development practice. Root-cause analysis helps route work to the appropriate team and supports durable corrective action.
Impact and severity assessment	Assess user impact and severity using the organization's established defect or risk model. Relevant considerations can include whether a task is blocked, importance of the affected task or content, affected user groups, availability and practicality of workarounds, frequency, scope, and applicable regulatory or contractual risk. Do not assume one universal severity scale.
Trend analysis	Comparing current findings with previous digital accessibility test reports to detect trends and frequently occurring digital accessibility defects across releases.

4.3.2 Typical Criteria Associated with Findings

Each finding reported should be documented with a consistent set of criteria to make it clear, understandable, and actionable.

Typical criteria for documenting a digital accessibility finding include:

Table 9: Typical Criteria for Documenting Digital Accessibility Findings

Title	A concise summary of the defect. (e.g., "Checkout button is not keyboard accessible").
Description	A clear explanation of the finding, the accessibility barrier or non-conformance, and the affected context.

© International Software Testing Qualifications Board

Environment	The OS, browser and assistive technology used during the test (including version numbers and any additional digital solution-related detail).
Steps to reproduce	A numbered list of steps that a developer can follow to replicate and observe the defect.
Evidence	Evidence sufficient to reproduce and understand the finding, such as screenshots, code or DOM excerpts, logs, screen-reader output, recordings, or other relevant artifacts. Evidence involving participants must follow applicable consent, privacy, and data-protection requirements.
Actual result	A description of the observed, incorrect behavior.
Expected result	A description of the correct behavior.
Applicable requirement / success criterion	Identify the applicable WCAG success criterion(s), other accessibility standard requirement(s), contractual criterion, or organizational requirement that the finding relates to. Not every accessibility finding is necessarily a WCAG violation.
Impact on users	A brief explanation of how this defect affects people with specific disabilities (e.g., "This prevents keyboard-only users and screen reader users from completing their purchase.").
Suggested remediation	Where appropriate, provide remediation guidance or references to verified patterns and techniques. The tester should not be required to prescribe a specific implementation when further design or development analysis is needed.

4.3.3 Best Practices in Reporting and Communicating Test Results

Effective communication is key to confirming that digital accessibility defects are understood and addressed. The goal is to be a collaborative partner, the enabler to overcome the defects, not just a defect finder.

A digital accessibility tester should be able to apply good reporting practices:

- Foster collaboration and empathy
- Manage defensive reactions with objective facts
- Adapt to the organization's digital accessibility maturity
- Tailor defect report to different audiences
- Be constructive and educational
- Use a combination of reporting formats

© International Software Testing Qualifications Board

- Focus on solutions, not just problems
- Report positive findings
- Track and measure progress

Foster collaboration and empathy

Involve cross-functional team members (developers, designers, product managers) directly in the test activities. Inviting them to observe live test sessions - especially end-user testing - is a highly effective way to build empathy and a shared understanding of the barriers users face. Make it easy for them to participate by organizing informal live observation sessions, scheduling them at convenient times, and emphasizing that even observing a partial session is valuable. Communicate clearly and tailor the terminology to the audience, avoiding overly dense digital accessibility jargon when discussing user impact with non-technical stakeholders.

Manage defensive reactions with objective facts

In situations involving conflict or pushback, the tester should rely on objective evidence and applicable requirements. When stakeholders have strong subjective opinions, shift the discussion toward reproducible observations, applicable WCAG success criteria or other requirements, and real-world feedback from users with disabilities.

Adapt to the organization's digital accessibility maturity

The W3C Accessibility Maturity Model [W3C AMM] is non-normative organizational guidance for assessing how accessibility is integrated into organizational processes. It can help a tester tailor communication to organizational maturity, but it is not a product-conformance standard.

Tailor defect report to different audiences

Provide developers with a detailed technical defect report and, where appropriate, suggestions or guidance for remediation. Identify the applicable accessibility requirement or WCAG success criterion when the finding relates to one, while recognizing that some user barriers or quality findings may fall outside WCAG. For product managers or executives, present a concise summary of risk, scope, priorities, and trends. For designers, emphasize early evaluation of visual and interaction patterns such as contrast, focus, structure, and layout.

Be constructive and educational

Beyond reporting findings, testers can provide educational context and point teams to reliable accessibility resources. Explain the user impact and the applicable requirement where relevant, while keeping responsibility for design and implementation decisions with the appropriate project roles.

Use a combination of reporting formats

There are different ways to report defects depending on the project phase and project need. The digital accessibility defects to be reported should use a dedicated defect management system. Log individual,

© International Software Testing Qualifications Board

actionable tickets with all the criteria listed in Section 4.3.2 above.

For formal reporting, prepare the type of report required by the context. An internal or project test report can summarize scope, methods, risks, findings, and remediation status. Where procurement or external conformance reporting requires an Accessibility Conformance Report (ACR), use the applicable VPAT® edition or other required format (see Section 4.3.4).

Focus on solutions, not just defects

Work with developers and designers to brainstorm and validate solutions. Help to perform confirmation testing for fixes and provide positive feedback when defects are resolved correctly. Product manager input is as important from the business side as the insights from developers and designers.

Report positive findings

Digital accessibility audits can often feel like a long list of failures for the development team. Always highlight features that work well in summaries and presentations (e.g., “The application's focus management and color contrast are excellent across all pages”). Including positive findings provides a balanced view, boosts team morale, and helps developers know which accessible features and patterns should not be modified or deleted.

Track and measure progress

Use dashboards to track the number of open versus closed digital accessibility defects over time. Monitor trends to see if the number of new defects being introduced is decreasing, indicating that the team is learning and improving.

4.3.4 The Purpose and Structure of a Formal Digital Accessibility Report

While defect management systems manage the day-to-day resolution of individual digital accessibility defects, formal digital accessibility test reports serve a broader purpose. They provide a comprehensive, point-in-time assessment of a product's conformance within a defined scope, against specified accessibility requirements. Such reports should identify the evaluated product/version, scope, applicable standard or guideline, methodology, sample, limitations, and date. These reports support communication of digital accessibility status to stakeholders, regulatory or procurement needs, and remediation planning.

Formal Reporting Formats: Internal versus External

Unlike day-to-day defect reporting, formal reports summarize a defined evaluation for broader stakeholders. The required report type depends on the purpose, such as internal remediation, governance, procurement, or external conformance reporting:

- Internal Digital Accessibility Evaluation Report. Created for product, engineering, testing, and governance stakeholders. It can summarize the evaluation scope and methods and reference

© International Software Testing Qualifications Board

detailed findings maintained in the organization's normal defect or work-management system rather than duplicating every defect in the report.
- Accessibility Conformance Report (ACR): An ACR documents how an ICT product or service conforms to specified accessibility standards or guidelines. The Information Technology Industry Council (ITI) publishes the Voluntary Product Accessibility Template (VPAT®), which is a template used to create an ACR. Different VPAT editions support reporting against WCAG, Section 508, EN 301 549, or combinations of these requirements. An ACR is not a certification and, by itself, is not proof of legal compliance [VPAT].

Typical ACR structure when using a VPAT®

A completed VPAT-based ACR typically identifies the product and version, report date, contact information, evaluation methods, applicable standards or guidelines, and conformance terminology, followed by requirement-by-requirement conformance information and remarks or explanations. The exact structure depends on the VPAT edition selected.

- Product and report identification. Identify the product or service, version, report date, and reporting organization.
- Applicable standards and scope. Identify the VPAT edition and the accessibility standards or guidelines being reported against, together with the scope of the evaluation.
- Evaluation methods. Describe the methods, tools, assistive technologies, and other evaluation approaches used to support the report.
- Contact and supporting information. Provide appropriate contact information and any notes needed to understand the evaluation context.
- Conformance terms. Use the conformance terminology defined by the selected VPAT edition consistently.
- Requirement-by-requirement results. Record the conformance level and explanatory remarks for each applicable criterion or requirement included in the selected VPAT edition.

© International Software Testing Qualifications Board

5. Assistive Technology – 75 minutes

Keywords

(none)

Domain-specific keywords

assistive technology, accessibility API, accessibility tree, screen magnifier, screen reader, voice control

Learning Objectives for Chapter 5:

5.1 Introduction to Assistive Technology

DAT-5.1.1 (K2) Explain assistive technologies

DAT-5.1.2 (K3) Apply assistive technologies in digital accessibility testing

5.1 Introduction to Assistive Technology

5.1.1 Understanding Assistive Technologies

To understand digital accessibility testing, one must first understand how users with disabilities interact with digital environments. AT refers to any piece of equipment, software program, or product system that is used to increase, maintain, or improve the functional capabilities of persons with disabilities.

For a user who cannot see a screen, cannot hear the audio, or cannot use the standard mouse and keyboard, assistive technology acts as a bridge. It translates the digital interface into a format the user can perceive (such as speech or magnification) and allows the user to send commands back to the system (such as voice commands or specialized keyboard inputs).

It is important to emphasize that assistive technology is not being tested; rather, the tester uses assistive technology to evaluate how the digital solution works in a particular configuration. For example, the goal is not to find defects in how a specific screen reader operates, but to verify that the software product exposes information and controls in a way the screen reader can interpret and operate.

Main categories of testing with assistive technology

Assistive technologies encompass a broad range of tools designed to support individuals in accessing digital content. Major categories include:

- Screen readers, which convert digital text, interface elements, and structural information into synthesized speech or output on a refreshable braille display. They are primarily used by people who are blind or have low vision.

© International Software Testing Qualifications Board

- **Screen magnifiers and display adjustments**, which enlarge content and enhance contrast. They are crucial for users with low vision.
- **Speech recognition** systems, enabled by voice control, allow navigation and text entry through spoken voice commands. Used by individuals with motor or mobility impairments.
- Alternative input devices include switch controls (one or more switches used to navigate and activate interface elements), eye-tracking systems, sip-and-puff devices (input controlled by air pressure), and adaptive keyboards. These can provide access for people with limited mobility.

Understanding these categories is essential because each represents distinct interaction patterns and digital accessibility needs across different hardware (desktop versus mobile).

The technical bridge between software and assistive technology

Assistive technologies obtain programmatic accessibility information from the browser, application framework, and/or operating system through platform accessibility APIs. These interfaces expose semantic information about user interface elements so assistive technologies can interpret and operate them.

Depending on the platform and technology stack, the browser, application framework, or operating system exposes an accessibility representation often described as an accessibility tree. This representation can include:

- role of a control (e.g., button, heading, checkbox),
- current state (e.g., selected, expanded, disabled),
- relationships between elements (e.g., label-input associations, table row-column structures),
- user-facing metadata (e.g., accessible names, descriptions, keyboard shortcuts).

Screen readers query this accessibility tree to render speech or braille. Similarly, screen magnifiers use the exposed properties to track keyboard/touch focus and caret position, while speech recognition software relies on predictable and accessible control names to map a user's voice command to the correct UI element.

The accuracy and completeness of this accessibility tree depend entirely on how well developers implement standard semantic markup. For web applications it means native HTML and ARIA attributes. For mobile and desktop applications this means utilizing UI components and digital accessibility properties specific to their frameworks (e.g., SwiftUI for iOS, Compose for Android, Windows Presentation Foundation (WPF) for Windows).

If an application fails to provide meaningful metadata, exposes custom widgets without digital accessibility support (like using a simple generic container to act as a button), or misuses semantic roles, the accessibility tree becomes incomplete or misleading. In such cases, assistive technologies may announce incorrect information, fail to expose essential controls, or prevent users from completing tasks altogether, creating critical digital accessibility barriers.

© International Software Testing Qualifications Board

5.1.2 Using Assistive Technologies

For digital accessibility testing, it is not sufficient to rely only on automated tools; testers should also use assistive technologies directly to evaluate how the product behaves in representative configurations. This can involve navigating a site or application with a screen reader to verify logical reading order and meaningful alternative text, using a screen magnifier to evaluate scaling without loss of functionality, or testing with voice control to determine whether interactive elements can be activated by spoken commands.

Through this manual testing, digital accessibility barriers become evident, such as unlabeled buttons, inaccessible dynamic content, or tasks that require complex mouse or touch gestures without accessible alternatives. To successfully apply assistive technology in testing, testers must use specific techniques tailored to the interaction model of the technology and the device being used.

When using assistive technology in testing, testers should structure their approach using the following methods.

Testing with screen readers

Testing with a screen reader involves more than simply listening to a page read from top to bottom. A tester must actively apply different navigation strategies to uncover structural and interactive defects across web, desktop, and mobile environments:

Testers may use assistive technology-specific commands (e.g., keyboard shortcuts on desktop or the “rotor/menu” touch gestures on mobile) to navigate the interface. The goal is to verify that the underlying semantic markup creates a logical document outline that matches the visual hierarchy.

When interacting with complex widgets (like tabs, accordions, or custom toggles), the tester must verify that the assistive technology announces not just the element's name, but its dynamic state (e.g., “Tab, Expanded” versus “Tab, Collapsed”) as it changes in real-time.

Some desktop screen readers use modes commonly described as browse/read mode and focus/forms mode, but names and switching behavior vary by product and configuration. Testers should verify that interactive widgets expose appropriate semantics and keyboard behavior so the target screen reader can be operated as intended, rather than assuming a specific mode-switching mechanism.

Testing with a screen magnifier

Zooming the interface to simulate low-vision usage requires evaluating how the software handles extreme scaling (often up to 400%):

When the screen is highly magnified, only a smaller part of the interface is visible. The tester must navigate via the keyboard or touchpad to determine if the magnifier automatically shifts/slides to keep the currently focused element visible on screen.

The tester verifies that tooltips, dropdowns, or sticky banners triggered by mouse hover or keyboard focus do not obscure the very same content the user is trying to magnify and read.

© International Software Testing Qualifications Board

Testing with voice control

Navigating the application using spoken commands requires verifying the mapping between visual design and programmed code.

“Label in Name” verification. Testers apply voice commands by speaking the visible text on the screen (e.g., saying “Tap Submit”). If the developer gave the button a different, hidden accessible name (e.g., a programmatic label of “Send Form”), the voice command will fail. The tester must practically verify that the visible text is always included within the programmatic name for digital accessibility.

Verifying platform-specific interactive behaviors

Native and well-supported platform controls generally expose expected roles, states, and interaction semantics through accessibility APIs. Custom controls require equivalent semantics and behavior to be implemented and tested for the target platform and assistive technology.

Interaction methods vary by operating system, browser, assistive technology, device, and widget pattern. Keyboard commands, modifier keys, screen-reader gestures, and control behavior should therefore be verified for the supported configuration rather than treated as universal.

The following table provides illustrative interaction examples only. Exact commands and gestures must be confirmed for the supported platform, assistive technology, and control pattern:

Table 10: Illustrative Assistive Technology Interaction Examples

Element / Action	Illustrative desktop interaction	Illustrative mobile screen-reader interaction	**Expected behavior to verify**
Navigate focusable items	TAB (or Shift + TAB to go backward)	Swipe Right (or Swipe Left to go backward)	Verify that focus/navigation order is logical, operable, and does not trap the user. Exact commands depend on the supported platform and assistive technology.
Activate a button	Enter (or sometimes Spacebar)	Double tap anywhere on the screen	Verify that the control can be activated and that resulting changes are exposed appropriately. Activation commands vary by control, platform, and assistive technology.

© International Software Testing Qualifications Board

Select a checkbox	Spacebar	Double tap	Verify that the control exposes and changes its selected/checked state using the supported interaction pattern.
Select a radio button	Arrow Keys (Up/Down/Left/Right)	Swipe to navigate, Double tap to select	Verify that radio-group navigation and selection follow the supported native or ARIA pattern for the target configuration.
Interact with a dropdown list	Down Arrow / Alt (or Option) + Down Arrow	Double tap to open, Swipe to browse items	Verify that the list or selection control can be opened, navigated, and operated using the supported interaction pattern for the target configuration.

By combining technical knowledge of assistive technology categories, awareness of how they integrate with platforms, and practical test techniques, software testers can more effectively identify and address digital accessibility defects, providing a more inclusive digital experience.

© International Software Testing Qualifications Board

6. Tools for Digital Accessibility Testing – 105 minutes

Keywords

static analysis, user experience

Domain-specific keywords

artificial intelligence, assistive technology, CI/CD pipeline, Document Object Model, linter, POUR, remediation

Learning Objectives for Chapter 6:

6.1 Types of Automation Tools for Digital Accessibility Testing

DAT-6.1.1 (K2) Summarize test automation and its limitations in digital accessibility testing

DAT-6.1.2 (K2) Explain the role and risks of artificial intelligence in digital accessibility testing

6.2 Tools for Digital Accessibility Testing

DAT-6.2.1 (K2) Classify the categories of tools available

DAT-6.2.2 (K2) Explain the use of Category 1: Static analysis and CI/CD pipeline tools

DAT-6.2.3 (K2) Explain the use of Category 2: Automated UI-based scanners and manual helpers

DAT-6.2.4 (K2) Explain the use of Category 3: Assistive technologies

DAT-6.2.5 (K2) Explain the use of Category 4: Cognitive and readability tools

6.1 Types of Automation Tools for Digital Accessibility Testing

Tools can improve the efficiency and repeatability of digital accessibility testing, but no single tool or tool category replaces human judgment, assistive-technology-supported testing, and, where appropriate, end-user involvement.

6.1.1 Test Automation and its Limitations in Digital Accessibility Testing

In digital accessibility, test automation tools programmatically evaluate source code, rendered user interfaces, or other artifacts against implemented accessibility rules. Their capabilities vary by tool, technology, ruleset, and available context.

Rule-based tools are effective where a condition can be evaluated programmatically, but many accessibility questions require contextual or human judgment. For example, a tool may detect whether an image has an alt attribute but may not reliably determine whether the alternative text communicates the appropriate meaning in context.

Main benefits

© International Software Testing Qualifications Board

The main benefits of test automation for digital accessibility:

- Efficiency and scale. Automated test tools can crawl thousands of pages or evaluate complex component libraries in a fraction of the time required for manual inspection, identifying rule-checkable issues (e.g., missing alternative text, contrast errors) in seconds.
- Consistency and repeatability. Test automation eliminates human fatigue, providing confidence that a missing <alt> text or invalid ARIA attribute is caught every time a scan is run, regardless of the tester's experience level or any other situational circumstances.
- Instant feedback. Integrated tools provide developers with “just-in-time” alerts, allowing defects to be resolved before the code is shared or integrated.

Capabilities versus limitations

To understand the limitations of automation, it is important to distinguish differences between “Technical conformance” (what a machine can see) and “Interaction capability” (what a human can experience).

Feature	What test automation can check	What human evaluation can assess
Images	Does the <img> tag have an alt attribute?	Is the description meaningful and helpful in context?
Forms	Is there a <label> associated with the <input>?	Does the label clearly describe the expected input?
Headings	Is the heading structure (h1-h6) numerically sequential?	Do the headings logically describe the content sections?
Keyboard	Is an element technically focusable in the code?	Is there a visible focus indicator? Is there a keyboard trap?
Color	Is the mathematical contrast ratio sufficient?	Does the design rely *only* on color to convey meaning?

Limitations of automation for digital accessibility:

- Coverage limitations. Automated tools evaluate only the accessibility conditions represented by their rules and available context. A report with no automated findings is therefore not evidence that a product is accessible or conformant.
- The noise of false-positive results. Automated tools may flag violations that are not actually barriers (e.g., reporting a contrast error on text that is positioned over a complex gradient or image background that the tool cannot parse). High rates of false positives can lead to “alert fatigue”, causing teams to ignore valid findings.

© International Software Testing Qualifications Board

- False sense of security. Stakeholders might misinterpret a high “digital accessibility score” from an automated tool as a guarantee of legal compliance, leading to the neglect of essential end-user testing and assistive technology validation. Significant barriers (like keyboard traps or illogical tab order) often go undetected by automation.

6.1.2 The Role and Risks of Artificial Intelligence in Digital Accessibility Testing

AI and machine learning can augment digital accessibility tools by assisting with pattern recognition, content analysis, prioritization, and remediation suggestions. AI-assisted results remain probabilistic and must be verified against the actual product behavior and applicable accessibility requirements.

The role of AI in digital accessibility testing is constantly increasing. AI tools can use visual analysis (“computer vision”) to detect defects that code-based tools miss, such as missing focus indicators, poor visual grouping, or content that overlaps when zoomed.

Large Language Models (LLMs) and Generative AI (GenAI) can evaluate page context to suggest meaningful alternative text for images or more descriptive labels for form fields. Furthermore, GenAI “auto-fix” capabilities can accelerate remediation by generating suggested code corrections for common violations. AI tools can also analyze complex text and suggest simpler alternatives, supporting the WCAG understandable principle. (Note: for a comprehensive understanding of testing AI systems and using AI in testing, please refer to the ISTQB Specialist Syllabus, Generative AI [ISTQB_AI_GEN] and AI Testing [ISTQB_AI_TEST]).

Risks and challenges of AI-driven tools:

Governance considerations include human review of AI-generated findings and fixes, protection of confidential or personal data, evaluation of bias and uncertainty, traceability of generated recommendations, and revalidation when models or prompts change.

Probabilistic versus deterministic results	While traditional automation is deterministic (if rule A is broken, then report B), AI is probabilistic, meaning it provides a “best guess” in a given situation. This can lead to unpredictable results or inconsistencies between digital accessibility scans.
Hallucinations and defects	AI may generate incorrect descriptions (e.g., describing a “submit” button as a “search” button) or provide remediation code that is technically invalid or introduces new digital accessibility barriers.
Data privacy	Using AI tools often requires sending code or content to a third-party model, which can raise significant security and data privacy concerns for an organization.
The “experience” gap	AI does not have lived experience of disability. AI-assisted accessibility findings and remediation suggestions require human verification and, where the evaluation objective includes user

© International Software Testing Qualifications Board

	experience, input from representative users.

6.2 Tools for Digital Accessibility Testing

To provide a comprehensive digital accessibility testing strategy, testers must be aware of a broad range of specialized software. While automation provides a baseline of efficiency, a professional toolkit incorporates a Layered Tooling Strategy. This involves using different types of tools to evaluate different layers of the application, from the raw source code and design tokens to the final rendered interface and the actual user experience as perceived through assistive technology.

This section defines these tools by their technical purpose and implementation within the SDLC.

6.2.1 Categories of Tools Available

To build a robust digital accessibility testing strategy, a tester must understand how to classify tools based on their technical capabilities rather than just their brand names. Classification is essential for identifying “coverage gaps” and helping prevent the team from over-relying on one type of testing activity while neglecting others.

Tools can be compared using three descriptive criteria — where they run, how they work, and who primarily uses them. These criteria describe tool characteristics; they are not the four syllabus categories introduced below.

Where does it run?	How does it work?	Who primarily uses it?
The environment	The evaluation method	The user persona
Tools range from source-code analyzers (linters) used by developers, to browser-based scanners used by Testers, and system-level software like screen readers that interact with the entire operating system.	This distinguishes between: ● Fully automated engines that follow binary rules; ● Guided tools that provide visual aids to help a human make a judgment call; ● Manual tools that require the tester to observe the interaction directly.	Tools are optimized for specific roles, including: ● Designer-focused: Used during the Requirements and Design phase to promote accessible foundations (e.g., contrast plugins, annotation tools). ● Developer-focused: Integrated into coding environments to support shift left technical fixes and automated remediation.

© International Software Testing Qualifications Board

		• Tester-focused: Designed for comprehensive audits, assistive technology validation, and defect management. • Content-focused: Focused on linguistic clarity and media digital accessibility (e.g., readability analyzers). • End-user-focused: While primarily assistive in nature (e.g., screen readers), these tools are the primary interface for users with disabilities and are utilized by testers to validate real-world interaction capability.

Note: While these are the primary technical criteria for classification, other factors such as cost, licensing, platform compatibility, and reporting features are also considered when selecting tools for an organization.

The four tool categories

For this syllabus, tools are grouped into four categories according to their primary technical environment and testing purpose. Using multiple categories can broaden coverage and reduce gaps, while recognizing that each category has limitations and that capabilities can overlap.

Table 11: Digital Accessibility Tool Categories

Category	Tools	Purpose	Level
Category 1	Static analysis and CI/CD pipeline tools	Check selected structural and source-level accessibility rules early	Code/source level
Category 2	Automated UI-based scanners and manual helpers	Evaluate selected accessibility properties of the rendered user	UI level

© International Software Testing Qualifications Board

		interface	
Category 3	Assistive technologies	Evaluate product behavior through representative assistive-technology configurations	Operational system level
Category 4	Cognitive and readability tools	Support analysis of content clarity, readability, and cognitive load	Content level

Mapping criteria and POUR principles to categories

While many digital accessibility tools identify defects that overlap across multiple areas, each category is defined by its primary technical environment and its core alignment with the POUR principles: (perceivable, operable, understandable, robust).

Table 12: Tool Categories and POUR Principle Alignment

Category	Primary environment	Evaluation method	Primary Persona	POUR principal alignment
Category 1: Static Analysis & CI/CD	source code / design	automated	developer / designer	robust (core), perceivable, operable
Category 2: UI Scanners & Helpers	rendered UI	automated / guided	tester / designer	perceivable (core), operable, robust
Category 3: Assistive Technology	operational system	manual	tester / end user	operable (core), perceivable, understandable
Category 4: Cognitive & Readability	content / UI	guided / automated	content editor	understandable (core)

Cross-category functionality and relationship to POUR principles:

The alignments above are not strictly separate. Digital accessibility testing is holistic, and tool categories frequently overlap in their coverage. Common examples of such overlap include:

© International Software Testing Qualifications Board

- Many-to-many relationships.
  A single tool in Category 2 (UI Scanner) might catch a missing alternative text (perceivable), a missing form label (operable), and a syntax error in ARIA code (robust).
- Categories 1 & 2 overlap.
  Both identify structural and technical defects. While Category 1 catches them in the source code, Category 2 catches the same defects in the rendered UI, often while simultaneously evaluating visual attributes.
- Category 3 as a cross-validator.
  While primarily used to test operability, assistive technology is essential for verifying perceivability (confirming content is actually announced or displayed) and understandability (verifying that announcements are logical and instructions are clear).
- Category 4 overlap.
  While focused on understandable content, readability tools often identify defects that require changes to the design (perceivable) or the technical structure of the text (robust) to improve clarity.

6.2.2 Category 1: Static Analysis and CI/CD Pipeline Tools

Category 1 tools operate at the source-code level or within the design environment. Their primary purpose is to verify that the technical foundation of the software is structurally robust before it is even rendered. By integrating these tests into the development workflow, teams can identify and resolve defects early, significantly reducing the cost and effort of remediation.

The strategic value of static analysis

The shift-left philosophy is the primary driver for Category 1 tools. The cost of fixing a digital accessibility barrier increases significantly as the project moves toward production. A missing form label identified by a linter takes seconds to fix. The same defect identified during an audit of a production site may require a full development sprint to resolve it.

Typical implementation methods

To align with the SDLC phases, Category 1 tools typically support two methods of implementation:

- IDE Extensions and Linters (real-time). These tools run inside the developer's Integrated Development Environment (IDE). They act as a “spell-checker” for digital accessibility, highlighting violations as the code is typed. This facilitates immediate correction during the Implementation Phase.
- CI/CD Pipeline Integration (Quality Gates). These tools run as part of the automated build process during the Build and Integration Phase. If a violation of a critical digital accessibility rule is detected, the build is configured to fail, preventing the code from being merged. This mechanism is known as a Quality Gate.

Example: Static analysis versus UI-based (dynamic) scanning

To understand the necessity of a layered tool strategy, a tester must distinguish between testing the “structural intent” (Category 1) and the “rendered outcome” (Category 2).

© International Software Testing Qualifications Board

Scenario: A page contains a form component whose final labels, states, and validation messages are partly generated at runtime.

Static analysis (Category 1) can evaluate source-code patterns before execution, such as whether required attributes or associations are present in the component source. UI-based scanning (Category 2) evaluates the rendered interface and can detect issues that depend on the runtime DOM, computed styles, generated states, or dynamic content.

The two categories therefore provide complementary evidence at different technical stages; neither establishes complete accessibility on its own.

Common technical validations and persona relevance

Category 1 tools serve both designers (using design-system plugins) and developers (using linters) by enforcing the “digital accessibility blueprint” before the UI is built:

- Structural Integrity: Verifying heading order and landmark presence (relevant to designers for layout logic and developers for semantic HTML).
- Attribute verification: Confirming images have alt attributes and form inputs have associated <label> tags (relevant to developers).
- ARIA validation: Evaluating roles and attributes against technical specifications (relevant to developers).
- Duplicate identifiers: Detecting duplicate <id> attributes that break assistive technology associations (relevant to developers)
- Visual baseline constraints: Assessing color contrast and font sizes within design systems before implementation (relevant to designers).

6.2.3 Category 2: Automated UI-based Scanners and Manual Helpers

Category 2 tools evaluate the rendered User Interface and detect digital accessibility defects that occur when the application is fully rendered and interactive. This is critical for the POUR perceivable principle, as many barriers only become detectable once the final functional interface is generated.

Capabilities of UI scanners

UI scanners evaluate the interface at runtime, analyzing the DOM in browsers or the view hierarchy in native applications. This run-time evaluation identifies:

- Dynamic UI defects - digital accessibility defects in interactive elements (e.g., modals, menus, or dynamic screen updates) that are generated during usage and are absent from the static source code.
- Visual attribute validation - actual color contrast ratio and text visibility within the rendered UI, accounting for transparency, background images, or overlapping layers.
- State-dependent violations - defects that occur during specific user interactions, such as error messages that are displayed only as a result of a failed input attempt.

Guided testing with manual helpers

Manual helpers are specialized tools that provide the tester with visual or technical data to support manual verification. They support human judgment rather than providing a binary pass/fail result.

© International Software Testing Qualifications Board

- Color contrast analyzers - tools used to calculate the contrast ratio between foreground and background colors to determine that text, icons, and interactive components remain clearly distinguishable. These tools allow testers to verify conformance either by manually entering numerical color codes (e.g., HEX or RGB values) or by using a “dropper” tool to sample colors directly from the rendered interface. This supports the POUR perceivable principle by verifying visual clarity against the actual background seen by the user.
- Visual inspection tools - they project technical metadata over the UI to assist the tester in verifying the heading hierarchy, the logical tab order, or the correct placement of digital accessibility landmarks.

Category 2 tools are commonly used as browser extensions, platform-specific inspectors, or automated scripts in development and staging environments. Their results provide preliminary evidence about detectable programmatic issues, but they do not establish compliance or complete conformance. Manual and assistive-technology testing should be planned according to risk, scope, and project maturity rather than waiting for every automated finding to be resolved.

6.2.4 Category 3: Assistive Technologies

AT refers to specialized software and hardware used to validate that a digital solution is operable and perceivable for users with disabilities. While automated tools evaluate technical code, assistive technology used in testing helps to verify the functional user experience.

The Digital Accessibility Tree

Automated scanners primarily evaluate the DOM or source code. In contrast, assistive technologies interact with the accessibility tree.

The accessibility tree is a specialized data structure generated by the browser or operating system from the underlying UI structure. It is a filtered version of the interface that contains only the information relevant to assistive technology, such as names, roles, states, and properties of interactive elements. Visual-only information, such as purely decorative images or layout styles, is omitted to provide an efficient experience for the user. Testing with assistive technology is required because browsers or operating systems may incorrectly map technical code to the digital accessibility layer, creating defects that automated scanners are unable to identify.

The following table explains the specific functional tests that testers perform using assistive technology:

Type of functional test	**Description**
Structural and non-visual navigation	Verifying that the interface remains navigable using semantic structures (e.g., landmarks, headings) and logical focus order without relying on visual context.
Interactive control identification	Confirming that interactive elements (buttons, links, form fields) correctly expose their name, role, and current state to the accessibility API for communication via assistive technology.
Dynamic notifications and	Confirming that background events, live updates, or programmatic state

© International Software Testing Qualifications Board

state changes	changes (e.g., a timer finishing, a file upload completion, or an element becoming expanded/collapsed) are correctly communicated to assistive technology users.
Input methods and alternative devices	Validating that all functionality is triggerable using various interaction modes and devices (e.g., switch controls, voice commands, and alternative keyboards) and determining there is no reliance on specific physical gestures or high-precision pointing.
Data entry and error handling	Verifying that transactional feedback, such as field labels, instructions, and error messages triggered by user input, are correctly associated and communicated via assistive technology so that users can complete tasks and recover from input errors.
Visual presentation and system settings	Verifying that the interface remains functional and readable when system-level digital accessibility settings are applied, such as high-contrast modes, large text scaling, or screen magnification.

AT supports the following areas:

- Visual support: screen readers (translating text to speech or Braille), refreshable Braille displays, and screen magnifiers.
- Motor support: switch Access (peripherals for step-by-step navigation), voice control software, and alternative keyboards.
- Auditory support: hearing aid compatibility tools, system-level visual notifications for sound, and software for managing captions/transcripts.
- Cognitive and learning support: reading assistants (to reduce visual clutter or highlight text), text-to-speech software (for auditory reinforcement of text), and simplified navigation interfaces.
- Speech support: augmentative and alternative communication (AAC) devices and software that allow users to produce speech via text or symbols.

A digital solution that conforms to automated tests but fails to function correctly when tested with Category 3 tools is considered *inaccessible*.

6.2.5 Category 4: Cognitive and Readability Tools

Category 4 tools can support analysis of content clarity and readability, which may contribute to the WCAG Understandable principle and broader cognitive accessibility goals. They do not, by themselves, determine cognitive accessibility or WCAG conformance.

Category 4 tools primarily assist content authors/editors in performing specific types of evaluations (linguistic and readability analysis) by utilizing linguistic metrics and automated evaluation methods. While these tools also provide value during the testing phase, they help to perform the following evaluations:

Content analysis metrics	**Description**

© International Software Testing Qualifications Board

Syntactic and lexical complexity analysis	Tools analyze sentence structure and word choice. They identify syntactic defects, such as overly long or nested sentences, and lexical defects, such as the use of high-level jargon instead of plain language, providing suggestions for simpler synonyms to reduce the cognitive effort required to process information.
Readability metrics	Readability formulas can provide indicators such as sentence or word complexity. A target reading level should be chosen for the intended audience, language, purpose, and applicable requirements; no universal grade-level target applies to all digital content.
Plain-language and figurative-language analysis	Tools can flag idioms, metaphors, ambiguous wording, or complex constructions for human review. Such results are indicators only and must be evaluated in context; machine-translation output should not be treated as proof that content creates a cognitive accessibility barrier.

Category 4 tools also help to perform the evaluation of the interface through content simplification and Sensory Support. These tools modify the visual or auditory presentation of content to help users with ADHD, autism, or dyslexia focus on information and reduce cognitive load without distraction:

Sensory and focus support	**Description**
Focus assistants	Software that strips away “visual noise” such as advertisements, navigation menus, and sidebars, presenting only the core text in a clean, high-contrast format.
Visual anchoring	Tools that guide the eye through gradients or by highlighting specific lines or parts of speech (e.g., nouns versus verbs) to improve reading comprehension and tracking.
Auditory reinforcement	“Read aloud” features that provide synchronized audio and visual tracking of text, supporting users who benefit from multi-sensory information delivery.

Consistency and predictability

To support the WCAG understandable principle, Category 4 tools assist in verifying the interface remains predictable across the entire application by helping to perform specific tests:

© International Software Testing Qualifications Board

- Terminology consistency. Confirming the same terms are used for the same actions (e.g., consistently using “Purchase” rather than switching between “Buy”, “Checkout”, and “Order”).
- Instructional clarity. Analyzing instructions to verify they do not rely on visual-only cues (e.g., “Press the circle to continue”), which can be confusing for those using cognitive support tools that modify layout.

Target Personas

While useful for testers, Category 4 tools are primarily optimized for Content Authors, UX Writers, and Marketing teams. By integrating these tools into the content creation process, organizations can establish that “clear writing” is treated as a fundamental digital accessibility requirement.

© International Software Testing Qualifications Board

7. References

7.1 Standards

The following standards are mentioned in these respective chapters.

- ISO/IEC 25010:2023 - Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Product quality model
- ISO 9241-171:2025 - Ergonomics of human-system interaction — Part 171: Software accessibility Chapter 1 and Chapter 2

7.2 ISTQB Documents

- [ISTQB_FL_SYL] ISTQB Foundation Level Syllabus, Version 4, 2024
- [ISTQB_AI_GEN] ISTQB Specialist Syllabus, Generative AI, Version 1.1, 2026
- [ISTQB_AI_TEST] ISTQB Specialist Syllabus, AI Testing, Version 2, 2026
- [ISTQB_UT] ISTQB Specialist Syllabus, Usability Testing, Version 2.0, 2026
- [ISTQB_UT_2018] ISTQB Specialist Syllabus, Usability Testing, Version 1.0, 2018

7.3 Books

- Heydon Pickering, “Inclusive Design Patterns”, 2016, Smashing Magazine GmbH, ISBN-13 9783945749463
- Dale Cruse, Denis Boudreau, “Inclusive Design for Accessibility: A practical guide to digital accessibility, UX, and inclusive web and app design”, 2025, Packt Publishing, ISBN-10 1835888232
- Reginé M. Gilbert , “Inclusive Design for a Digital World: Designing with Accessibility in Mind”, 2025, Apress, ISBN-13 979-8868818196
- Adam Silver, “Form Design Patterns”, 2018, Smashing Magazine, ISBN-13 978-3945749753
- Eric Meyer, Sara Wachter-Boettcher, “Design for Real Life” by Eric Meyer and Sara Wachter-Boettcher, 2016, Active Voice, LLC, ISBN-13 978-1952616372
- Kat Holmes, “Mismatch: How Inclusion Shapes Design”, 2020, The MIT Press, ISBN-13 978-0262539487

7.4 Other References

The following official sources provide current information about standards, guidelines, legislation, and related resources referenced in this syllabus. Standards and legal requirements may be revised after publication of this syllabus; readers should consult the responsible standards body or government source to determine the currently applicable version or requirements. External source content is not examinable beyond what is explicitly summarized in this syllabus.

© International Software Testing Qualifications Board

Official update and reference sources

[W3C ARIA]: https://www.w3.org/WAI/standards-guidelines/aria/
[W3C WCAG]: https://www.w3.org/WAI/standards-guidelines/wcag/
[W3C AMM]: https://www.w3.org/TR/maturity-model/
[VPAT]: https://www.itic.org/policy/accessibility/vpat/
[W3C WCAG2ICT]: https://www.w3.org/TR/wcag2ict-22/

[W3C WCAG-EM]: https://www.w3.org/WAI/test-evaluate/conformance/wcag-em/

[ISO 25010]: https://www.iso.org/standard/78176.html

[ISO 9241-171]: https://www.iso.org/standard/86308.html

[ETSI EN 301 549]: https://www.etsi.org/deliver/etsi_en/301500_301599/301549/03.02.01_60/en_301549v030201p.pdf

[EU WAD]: https://eur-lex.europa.eu/eli/dir/2016/2102/oj/eng

[EU EAA]: https://eur-lex.europa.eu/eli/dir/2019/882/oj/eng

[UK Accessibility]: https://www.gov.uk/guidance/accessibility-requirements-for-public-sector-websites-and-apps

[US ADA]: https://www.ada.gov/resources/web-rule/

[US Section 508]: https://www.section508.gov/develop/applicability-conformance/

[W3C Evaluation Tools]: https://www.w3.org/WAI/test-evaluate/tools/list/

© International Software Testing Qualifications Board

8. Appendix A – Learning Objectives/Cognitive Level of Knowledge

The specific learning objectives applying to this syllabus are shown at the beginning of each chapter. Each topic in the syllabus will be examined according to the learning objective for it.

The learning objectives begin with an action verb corresponding to its cognitive level of knowledge as listed below.

Level 1: Remember (K1)

The candidate will remember, recognize and recall a term or concept.

Action verbs: Recall, recognize.

Examples
Recall the concepts of the test pyramid.
Recognize the typical objectives of testing.

Level 2: Understand (K2)

The candidate can select the reasons or explanations for statements related to the topic, and can summarize, compare, classify and give examples for the testing concept.

Action verbs: Classify, compare, differentiate, distinguish, explain, give examples, interpret, summarize

Examples	**Notes**
Classify test tools according to their purpose and the test activities they support.	
Compare the different test levels.	Can be used to look for similarities, differences or both.
Differentiate testing from debugging.	Looks for differences between concepts.
Distinguish between project and product risks.	Allows two (or more) concepts to be separately classified.
Explain the impact of context on the test process.	
Give examples of why testing is necessary.	

© International Software Testing Qualifications Board

Examples	Notes
Infer the root cause of defects from a given profile of failures.	
Summarize the activities of the work product review process.	

Level 3: Apply (K3)

The candidate can carry out a procedure when confronted with a familiar task, or select the correct procedure and apply it to a given context.

Action verbs: Apply, implement, prepare, use

Examples	Notes
Apply boundary value analysis to derive test cases from given requirements.	Should refer to a procedure / technique / process etc.
Implement metrics collection methods to support technical and management requirements.	
Prepare installability tests for mobile apps.	
Use traceability to monitor test progress for completeness and consistency with the test objectives, test strategy, and test plan.	Could be used in a LO that wants the candidate to be able to use a technique or procedure. Similar to 'apply'.

Level 4: Analyze (K4)

The candidate can separate information related to a procedure or technique into its constituent parts for better understanding and can distinguish between facts and inferences. Typical application is to analyze a document, software or project situation and propose appropriate actions to solve a problem or task.

Action verbs: Analyze, deconstruct, outline, prioritize, select.

Examples	Notes
Analyze a given project situation to determine which black-box or experience-based test techniques should be applied to achieve specific goals.	Examinable only in combination with a measurable goal of the analysis. Should be of form 'Analyze xxxx to xxxx' (or similar).

© International Software Testing Qualifications Board

Examples	Notes
Prioritize test cases in a given test suite for execution based on the related product risks.	
Select the appropriate test levels and test types to verify a given set of requirements.	Needed where the selection requires analysis.

© International Software Testing Qualifications Board

9. Appendix B – Business Outcomes Traceability Matrix with Learning Objectives

This section lists the traceability between the Business Outcomes and the Learning Objectives of Digital Accessibility Testing syllabus

Business Outcomes: Digital Accessibility Testing			BO1	BO2	BO3	BO4	BO5	BO6	…
DAT-BO1	Explain the value of digital accessibility principles and their impact on users.		6 LOs						
DAT-BO2	Interpret global guidelines to apply digital accessibility standards.			5 LOs					
DAT-BO3	Define digital accessibility testing activities				8 LOs				
DAT-BO4	Identify, analyze, and report digital accessibility defects					7 LOs			
DAT-BO5	Use assistive technology for digital accessibility testing						2 LOs		
DAT-BO6	Select and use tools for digital accessibility testing							7 LOs	

© International Software Testing Qualifications Board

Business Outcomes: Digital Accessibility Testing			BO1	BO2	BO3	BO4	BO5	BO6	…
Chapter 1	**Basic Concepts**								
1.1	Fundamentals of Digital Accessibility								
DAT-1.1.1	Explain the concept of digital accessibility	K2	X						
DAT-1.1.2	Explain the objectives of a digital accessibility evaluation	K2	X						
DAT-1.1.3	Summarize key principles and practices for digital accessibility testing	K2	X						
1.2	Disabilities								
DAT-1.2.1	Explain the different categories of disabilities	K2	X						
DAT-1.2.2	Explain the importance of accessible software	K2	X						
1.3	Risks of Inadequate Digital Accessibility								
DAT-1.3.1	Summarize the risks and impact of not applying digital accessibility to software	K2	X						

© International Software Testing Qualifications Board

Business Outcomes: Digital Accessibility Testing			BO1	BO2	BO3	BO4	BO5	BO6	…
Chapter 2	**Digital Accessibility Standards**								
2.1	Digital Accessibility Guidelines								
DAT-2.1.1	Explain the general content and applicability of global guidelines to particular types of projects/applications	K2		X					
2.2	Digital Accessibility Regulations and Standards								
DAT-2.2.1	Explain standards and legislation	K2		X					
DAT-2.2.2	Explain the general content and applicability of the ISO 9241-171 Standard	K2		X					
DAT-2.2.3	Explain the structure of the Web Content Accessibility Guidelines and other applicable guidelines for software digital accessibility	K2		X					
DAT-2.2.4	Explain typical criteria described in the Web Content Accessibility Guidelines	K2		X					
Chapter 3	**Digital Accessibility Testing Activities**								

© International Software Testing Qualifications Board

Business Outcomes: Digital Accessibility Testing			BO1	BO2	BO3	BO4	BO5	BO6	…
3.1	The Approach to Digital Accessibility Testing								
DAT-3.1.1	Explain the planning activities for establishing a digital accessibility test	K2			X				
3.2	Digital Accessibility Testing Activities								
DAT-3.2.1	Explain the benefits of shift left in digital accessibility testing	K2			X				
DAT-3.2.2	Explain how to perform digital accessibility testing	K2			X				
DAT-3.2.3	Summarize the process of digital accessibility audits as a method for evaluating conformance	K2			X				
DAT-3.2.4	Explain the relevance of end-user testing in evaluating real-world digital accessibility	K2			X				
3.3	A Step-By-Step Approach to End-User Testing								
DAT-3.3.1	Explain the purpose and approach adopted in end-user testing	K2			X				

© International Software Testing Qualifications Board

Business Outcomes: Digital Accessibility Testing			BO1	BO2	BO3	BO4	BO5	BO6	…
3.4	Risks and Defects in Digital Accessibility Testing								
DAT-3.4.1	Explain the typical project risks and mitigation measures in digital accessibility testing	K2			X				
DAT-3.4.2	Explain common software defects identified during digital accessibility testing	K2			X				
Chapter 4	**Digital Accessibility Testing Lifecycle**								
4.1	Preparing a Digital Accessibility Test								
DAT-4.1.1	Explain factors to consider in digital accessibility test planning	K2				X			
DAT-4.1.2	Explain the configuration requirements for digital accessibility tests	K2				X			
4.2	Executing Digital Accessibility Tests								
DAT-4.2.1	Explain the applicable digital accessibility test tasks	K2				X			

© International Software Testing Qualifications Board

Business Outcomes: Digital Accessibility Testing			BO1	BO2	BO3	BO4	BO5	BO6	…
4.3	Analyzing and Communicating Findings								
DAT-4.3.1	Analyze digital accessibility test results to identify patterns, root causes, user impact, and priorities for action	K4				X			
DAT-4.3.2	Explain typical criteria associated with findings	K2				X			
DAT-4.3.3	Apply good practices to report and communicate digital accessibility test results	K3				X			
DAT-4.3.4	Summarize the purpose and structure of a formal digital accessibility report	K2				X			
Chapter 5	**Assistive Technology**								
5.1	Introduction to Assistive Technology								
DAT-5.1.1	Explain assistive technologies	K2					X		
DAT-5.1.2	Apply assistive technologies in digital accessibility testing	K3					X		

© International Software Testing Qualifications Board

Business Outcomes: Digital Accessibility Testing			BO1	BO2	BO3	BO4	BO5	BO6	…
Chapter 6	**Tools for Digital Accessibility Testing**								
6.1	Types of Automation Tools for Digital Accessibility Testing								
DAT-6.1.1	Summarize test automation and its limitations in digital accessibility testing	K2						X	
DAT-6.1.2	Explain the role and risks of artificial intelligence in digital accessibility testing	K2						X	
6.2	Tools for Digital Accessibility Testing								
DAT-6.2.1	Classify the categories of tools available	K2						X	
DAT-6.2.2	Explain the use of Category 1: Static analysis and CI/CD pipeline tools	K2						X	
DAT-6.2.3	Explain the use of Category 2: Automated UI-based scanners and manual helpers	K2						X	
DAT-6.2.4	Explain the use of Category 3: Assistive technologies	K2						X	

© International Software Testing Qualifications Board

Business Outcomes: Digital Accessibility Testing			BO1	BO2	BO3	BO4	BO5	BO6	…
DAT-6.2.5	Explain the use of Category 4: Cognitive and readability tools	K2						X	

© International Software Testing Qualifications Board

10. Appendix C – Release Notes

ISTQB® Digital Accessibility Testing Syllabus V1.0 is a new Specialist Level syllabus released in 2026. Principal characteristics of this release include:

Content lineage: This syllabus incorporates and expands digital accessibility testing topics that were previously included in the ISTQB® Specialist Syllabus, Usability Testing, Version 1.0 (2018). Those topics have been moved into CT-DAT v1.0 to provide a dedicated qualification focused on digital accessibility testing.

Six examinable chapters covering basic concepts, standards and legislation, digital accessibility testing activities, the testing lifecycle, assistive technology, and testing tools.

A minimum accredited-training duration of 675 minutes (11.25 hours), aligned with the approved Business Outcomes and Learning Objectives.

A standards baseline that includes ISO 9241-171:2025, WCAG 2.2, WCAG2ICT guidance, EN 301 549, and representative EU, UK, and U.S. accessibility requirements.

A layered testing approach combining automated checks, manual testing, assistive-technology testing, scoped conformance evaluation, and end-user testing.

Practical guidance for analyzing, prioritizing, reporting, and communicating digital accessibility findings, including use of Accessibility Conformance Reports (ACRs).

Coverage of current tool categories and the role, limitations, and risks of artificial intelligence in digital accessibility testing.

© International Software Testing Qualifications Board

11. Appendix D – Domain-Specific Keywords

Term Name	Definition
accessibility API	A platform interface exposed by a browser, application framework, or operating system that provides programmatic accessibility information to assistive technologies.
accessibility conformance report	A report that documents how an information and communication technology product or service conforms to specified accessibility standards or guidelines. An ACR can be created by completing an applicable VPAT® template.
accessibility tree	A programmatic representation of user interface semantics, such as names, roles, states, properties, and relationships, exposed for use by assistive technologies. Its implementation depends on the browser, application framework, and operating system.
Accessible Rich Internet Applications (ARIA) [W3C ARIA]	A World Wide Web Consortium technical specification that defines how to make dynamic web content and advanced user interface controls more accessible to people with disabilities through semantic markup attributes
artificial intelligence (AI)	The capability of an engineered system to acquire, process, create and apply knowledge and skills
ARIA Authoring Practices Guide (APG)	A W3C resource providing design patterns, examples, and guidance for implementing accessible web interface patterns with ARIA.
assistive technology (AT)	A piece of equipment, component or system that is used to increase, maintain, or improve the functional capabilities of persons with disabilities.
CI/CD pipeline	The automated series of steps to deliver a new software version.
document object model (DOM)	A programming interface for web documents that represents the page structure.
linter	A static analysis tool used in digital accessibility to analyze source code integrated into the developer's environment to provide real-time feedback and prevent structural accessibility defects.
medical model of disability	A paradigm that views disability as an issue that exists within the person, caused by injury, illness, or a health condition, with the aim of diagnosing, treating, managing, or “fixing” the individual
POUR	An acronym representing the four foundational principles of the Web Content Accessibility Guidelines, which state that digital

© International Software Testing Qualifications Board

Term Name	Definition
	content must be perceivable, operable, understandable, and robust.
remediation	The process of correcting accessibility barriers or non-conformities in the code, content, design, or configuration of a digital component or system.
screen magnifier	A type of assistive technology that enlarges on-screen content and enhances visual contrast, crucial for validating the accessibility experience for users with low vision.
screen reader	A type of assistive technology that allows users to operate software without needing to view the visual display by converting digital text, interface elements, and structural code into synthesized speech or output on a refreshable braille display.
semantic markup	The use of HTML markup to reinforce the meaning and structure of the information in web pages, rather than merely to define its visual presentation, facilitates native compatibility with assistive technologies.
social model of disability	A paradigm that states that people are disabled by the physical, attitudinal, and systemic barriers that prevent them from participating equally because of society.
speech recognition	A type of assistive technology that enables interface navigation and text entry through spoken voice commands, used primarily by individuals with motor or mobility impairments.
success criterion	A testable statement within the Web Content Accessibility Guidelines that defines a specific requirement for digital accessibility conformance.
switch access	A type of assistive technology that allows users with significant motor impairments to navigate and interact with digital interfaces using one or more physical switches instead of a traditional mouse or keyboard.
voice control	A method of interacting with a digital interface using spoken commands.
voluntary product accessibility template [VPAT]	A template published by the Information Technology Industry Council (ITI) for creating an Accessibility Conformance Report (ACR) that reports conformance to selected accessibility standards or guidelines.
Web Accessibility Initiative (WAI)	A project to develop international standards, guidelines, and educational resources to make the internet universally accessible and inclusive for people with all types of disabilities.

© International Software Testing Qualifications Board

Term Name	Definition
Web Content Accessibility Guidelines [WCAG]	A set of accessibility recommendations developed by the World Wide Web Consortium.
World Wide Web Consortium (W3C)	The primary international standards organization for the World Wide Web.

© International Software Testing Qualifications Board

12. Index

A

accessibility 14
accessibility API 50, 52, 62
accessibility conformance report (ACR) 47, 48
accessibility evaluation 15, 17, 21, 35, 36, 41
accessibility tree 50, 62
Accessible Rich Internet Applications (ARIA) 32, 36, 44, 50, 53, 55, 61
ACR ..... See accessibility conformance report (ACR)
ADA ..... See Americans with Disabilities Act (ADA)
AI ..... See artificial intelligence (AI)
Americans with Disabilities Act (ADA) 24, 31
APG ..... See ARIA Authoring Practices Guide (APG)
ARIA ..... See Accessible Rich Internet Applications (ARIA)
ARIA Authoring Practices Guide (APG) 44
artificial intelligence (AI) 32, 56
assistive technology (AT) 18, 20, 21, 27, 30, 31, 33, 34, 35, 39, 40, 41, 43, 45, 48, 49, 51, 57, 61, 62
AT ..... See assistive technology (AT)
audit ..... See digital accessibility audit

C

CI/CD pipeline 31, 39, 42, 58, 60
compliance 21, 31, 40, 48, 62

D

digital accessibility 14
digital accessibility audit 22, 29, 30, 32, 33, 40, 47, 60
digital accessibility remediation ..... See remediation
document object model (DOM) 42, 45, 61, 62
DOM ..... See document object model (DOM)

E

EAA ..... See European Accessibility Act (EAA)
EN 301 549 ..... 23, 39, 48
end-user ..... See end-user testing
end-user testing 15, 29, 33, 34, 41, 42, 46
Equality Act 24
European Accessibility Act (EAA) 23, 31

I

interaction capability 14, 15, 17, 21, 55, 58
ISO 9241-171 24

L

linter 30, 42, 57, 60

M

medical model of disability 19

P

POUR 14, 25, 59, 61

R

remediation 32, 34, 36, 40, 43, 44, 45, 46, 47, 56, 57, 60

S

screen magnifier 32, 39, 50, 51, 63
screen reader 15, 18, 20, 32, 35, 37, 39, 45, 49, 51, 57, 63
Section 508 ..... 24, 39, 48
semantic markup 18, 50, 51
shift left 29, 30, 36, 41, 57, 60
social model of disability 19
speech recognition 32, 50
static analysis 42, 58, 60
success criterion 24, 25, 33, 45, 46
switch access 63

T

test planning 29, 38

© International Software Testing Qualifications Board

U

usability 14, 15, 17
user experience (UX) . 14, 15, 21, 31, 33, 34, 40, 57, 62, 65
UX .....See user experience (UX)

V

voice control 39, 50, 51, 63
voluntary product accessibility template (VPAT) 47, 48
VPAT ..... See voluntary product accessibility template (VPAT)

W

W3C ... See World Wide Web Consortium (W3C)
WAI .....See Web Accessibility Initiative (WAI)
WCAG .....See Web Content Accessibility Guidelines (WCAG)
Web Accessibility Directive 23
Web Accessibility Initiative (WAI) 25, 44
Web Content Accessibility Guidelines (WCAG) ...14, 16, 21, 23, 24, 25, 32, 39, 43, 45, 46, 48, 56, 63
World Wide Web Consortium (W3C) .. 14, 23, 25, 32, 36, 46

© International Software Testing Qualifications Board