

Sample Playbook for an Accessible SDLC

A starting point for building your own Accessibility Playbook for your Software Development Life Cycle.

Table of Contents

Sample Document - Playbook for an Accessible SDLC	1
Stage Gates	1
Ceremonies.....	3
Artifacts	6
Tools	15
Roles, Personas, and Ownerships	17

Stage Gates

Where the SDLC pauses to ensure accessibility has been included.

Stage Gate 1 – Discovery Readiness

When: Problem definition / early planning

Owner: Product +/- Design

A11y What: Target users & use cases defined; a11y risks identified; constraints & assumptions doc'd

Evidence: Problem statement(s); early reqs & notes

Signal to proceed: “We understand who might be excluded if we get this wrong.”

Stage Gate 2 – Design Readiness

When: Before detailed design

Owner: Design

A11y What: Accessible patterns selected and/or ID'd; content structure defined; interaction complexity understood; project roles outlined

Evidence: Flows / low-fi designs; pattern references; Design brief; personas ready / ID'd; testing strategy / knowledge gaps id'd

Signal to proceed: “Accessibility interaction plans noted”

Stage Gate 3 – Design Review

When: Hi-fi design complete

Owner: Design + A11y Champion w/ design

A11y What: Color contrast verified; states, focus, errors defined; responsive / zoom considered;

Evidence: Annotated designs; design evaluation notes

Signal to proceed: “Engineering won’t have to guess.”, “Designs are signed off as approved by XYZ”

Stage Gate 4 – Dev Readiness

When: Backlog grooming / sprint planning

Owner: Engineering + Design +/- Product

A11y What: A11y acceptance criteria defined; component behavior explicit; known risks acknowledged; test plans id’d; software tools installed;

Evidence: Requirements added to stories / tasks; acceptance criteria; linked designs; all interactions for new components have understood references;

Signal to proceed: “This is ready to build.” “Accessibility compliance scoped”

Stage Gate 5 – QA Testing

When: Feature complete / pre-release

Owner: QA + A11y Champion(s) +/- Engineering

A11y What: Keyboard flow validated; SR basics checked; critical issues resolved or tracked, Use Case testing, 3rd Party evaluation in dev/staging; support needs outlined;

Evidence: Test notes; defects / tickets; evaluation results.

Signal to proceed: “We know what we’re shipping.”

Stage Gate 6 – Documentation / Training

When: Code Freeze / Post-implementation / before scale

Owner: Design System / Enablement / Training

A11y What: Patterns documented; usage guidance added; limitations noted; external documents updated; documents authored and tested for accessibility.

Evidence: Design system updates/backlogged; internal & external documents noted; documents.

Signal to proceed: “This won’t break the next time it’s reused.” “Our teams and tools are ready to support our customers.”

Stage Gate 7 – Release & Risk Acceptance

When: Production release

Owner: Product + Leadership

A11y What: Residual risks understood; decisions explicit; follow-ups planned; production site test plan; ACR/VPAT planned; automated scans updated

Evidence: Release notes; risk items / backlog; external

Signal to proceed: “Risk is known and accepted.”

Stage Gate 8 – Retro / System Improvement

When: Post-release / retro

Owner: Full SDLC team; Scrum Master; Leadership

A11y What: Friction identified; upstream fixes defined; system gaps addressed; processes analyzed; ceremonies start/stop/continue;

Evidence: Retro notes; process or system updates; a11y score change’

Signal to proceed: “The system improves, not just the feature.”

Ceremonies

Where the activities or work happens and accessibility is practiced. Ensure the right people are present, and you have the Accessibility Champions bringing their expertise into the process.

Stand-Up

When: Daily

Owner: Team leads, scrum masters, project managers, product owners, etc.

A11y Requirement: One person needs to be an accessibility advocate to ask questions and keep people true to the stage gates.

Example from a design-only standup: “Before you start on the high-fidelity designs, do you mind if I check your personas for accessibility mapping?”

A11y What: A11y blockers, risks, unknowns surfaced. Stage gate reminders. Call-to-action and advocacy.

Evidence: Verbal updates; follow-up tickets and comments; sprint or story adjustments;

Backlog Grooming / Refinement

When: Ongoing / pre-sprint

Owner: Product owners, engineering leads, design leads

A11y Requirement: A11y expectations must be explicit before work is accepted into a sprint. Examples (product + dev grooming): “Do these acceptance criteria cover keyboard and error states, or do we need to add them?” “What accessibility issues have been reported and when are we planning them?”

A11y What: Testable A11y requirements; states and interactions clarified; risks acknowledged.

Evidence: Updated user stories; acceptance criteria; linked designs or notes.

Design Review / Critique

When: Design milestones

Owner: Design leads, designers, accessibility partners, accessibility SME

A11y Requirement: Design intent must be reviewable for accessibility before handoff. One or more people in the design review need to be the Accessibility champion. Designs need annotations, documentation, and/or resources.

Example (cross-functional review): “Can you walk me through the focus order and error handling for this flow?”, “The white on black is compliant, but it’s very intents and may introduce reading fatigue, can we use a softer tone from the design system?”, “I haven’t read your keyboard documentation, did you include a hidden skip link in the new nav redesign?”

A11y What: Contrast, focus, states, zoom, responsive behavior reviewed and validated.

Evidence: Annotated designs; review comments; updated design files.

Sprint Planning

When: Start of sprint

Owner: Product owners, engineering leads, scrum masters

A11y Requirement: A11y effort and risk must be understood before committing work. New barriers are addressed immediately. New components require more time for usability testing. Verification design has delivered the required documentation.

Example (dev planning):

“Is there anything in this story that changes how long accessibility will take to implement?”

A11y What: A11y scope acknowledged; implementation risks identified; estimates adjusted.

Evidence: Sprint backlog; estimation notes; scoped stories.

Development Sync / Engineering Stand-Up

When: Daily or as needed

Owner: Engineering leads, developers, QA, design as-needed, design system engineers

A11y Requirement: Implementation decisions must surface a11y tradeoffs early.

Example (dev-only sync):

“Are we using native semantics here, or do we need ARIA support? How does this work with the shadow DOM?”

A11y What: Keyboard handling, semantics, ARIA usage, implementation blockers discussed.

Evidence: PR comments; code notes; task updates.

QA / Testing Review

When: Feature complete / pre-release

Owner: QA leads, accessibility testers, engineers

A11y Requirement: A11y validation must happen before release decisions.

Example (QA handoff):

“Have we validated keyboard flow and screen reader output for the critical paths?”

A11y What: Keyboard, SR basics, contrast issues verified; defects logged or triaged.

Evidence: Test results; defect tickets; known-issue lists.

Sprint Review / Demo

When: End of sprint

Owner: Team + stakeholders

A11y Requirement: Shipped behavior must be demonstrable, not assumed.

Example (demo):

“Can we quickly tab through this flow so everyone can see how it behaves?”

A11y What: End-to-end accessibility behavior validated; gaps surfaced.

Evidence: Demo feedback; follow-up tickets; backlog updates.

Retrospective

When: End of sprint / release

Owner: Team

A11y Requirement: A11y friction feeds system improvement, not blame.

Example (retro):

“What accessibility issue could we have caught earlier?”

A11y What: Process gaps identified; upstream improvements proposed.

Evidence: Retro notes; action items; updated processes or stage gates.

Design System / Enablement Review (if applicable)

When: System changes or reuse moments

Owner: Design system teams, enablement, accessibility

A11y Requirement: Reusable patterns must meet baseline a11y expectations.

Example:

“Should this guidance live in the design system so teams don’t solve it again?”

A11y What: Defaults reviewed; documentation updated; guidance clarified.

Evidence: System updates; docs; release notes.

Design Hand-Off

When: Design complete / before development starts

Owner: Design leads, designers, engineering leads

A11y Requirement: Accessibility intent must survive the handoff.

Example (design -> dev):

“Here’s where focus should land after submit, and these are the required error announcements.”

A11y What: A11y requirements transferred; states, focus, interactions clarified; risks flagged.

Evidence: Annotated designs; handoff notes; linked tickets or stories.

Design Team Workshops

When: Early design exploration / problem framing

Owner: Design leads, designers, facilitators

A11y Requirement: Accessibility considered during ideation, not after.

Example (workshop prompt):

“How would someone using only a keyboard complete this flow?”

A11y What: Inclusive assumptions challenged; risky concepts identified early.

Evidence: Workshop notes; whiteboards; early concepts or flows.

Design Research Sessions

When: Discovery, validation, usability testing

Owner: Researchers, design, product

A11y Requirement: Research includes accessibility considerations and participants.

Example (research planning):

“Are we capturing accessibility needs or assistive tech usage in this study?”

A11y What: User needs validated; a11y insights captured; assumptions corrected.

Evidence: Research plans; findings; synthesis notes.

Regression Testing

When: Ongoing / before release

Owner: QA, accessibility, engineering

A11y Requirement: Existing accessibility behavior must not break.

Example (test review):

“Did this change affect keyboard flow or screen reader output elsewhere?”

A11y What: Regressions detected; baseline preserved; issues caught early.

Evidence: Regression test results; defect tickets; test coverage updates.

Project / Initiative Kickoff

When: Formal project start

Owner: Product, project management, engineering, design

A11y Requirement: Accessibility scope and accountability defined from day one.

Example: “Who owns accessibility validation and what stage gates apply to this work?”

A11y What: Roles clarified; expectations set; documentation requirements aligned.

Evidence: RACI; kickoff deck; project charter notes.

Artifacts

The elements that result from Ceremonies that should be utilized in other stages or cycles. In a completed Playbook, each artifact would be linked to a “how-to” guide with an example. For each artifact, each artifact should be reviewed for inclusive principles. Considerations include:

- What tools are involved with creating & maintaining these artifacts?
- Has an a11y champion reviewed the outcomes and given feedback on
 - o Making the artifacts themselves accessible
 - o Ensuring accessibility is considered within the artifact creation
- Have we utilized each artifact properly to ensure accessibility is ready at the next phase of our SDLC
- Are our artifacts themselves accessible for our internal stakeholders? (could an employee with a disability review our artifact?)

Discovery

Discovery Kickoff

Artifacts

- Discovery brief with accessibility risks noted
- Persona accessibility mapping
- Risk register entry
- Assumption log including disability impact
- Regulatory applicability notes
- Initial accessibility hypothesis notes
- Accessibility opportunity statements

Project / Initiative Kickoff

Artifacts

- Project charter with accessibility scope
- RACI including accessibility ownership
- Stage gate checklist reference
- Definition of Done including accessibility

- Accessibility success criteria summary
- Accessibility escalation path documentation

Feature Kickoff

Artifacts

- Feature brief with accessibility considerations
- Initial interaction map
- Risk annotation in story
- Accessibility complexity assessment
- Accessibility requirements summary

Design Research Sessions

Artifacts

- Research plan including accessibility recruitment
- Screener questions capturing assistive tech use
- Session guide with accessibility probes
- Research findings highlighting accessibility insights
- Assistive technology usage findings
- Inclusive persona updates

Planning Phase

Backlog Grooming / Refinement

Artifacts

- User stories with explicit accessibility acceptance criteria
- State and interaction checklist
- Linked design references
- Accessibility risk tags or labels
- Accessibility subtasks attached to stories

Sprint Planning

Artifacts

- Effort estimates including accessibility complexity
- Sprint backlog reflecting accessibility tasks
- Technical spike tickets if needed
- Implementation notes attached to stories
- Accessibility test task assignments

Roadmap / PI Planning

Artifacts

- Accessibility debt inventory
- Prioritized remediation roadmap
- Investment allocation notes
- Risk heatmap
- Accessibility maturity roadmap

Design Phase

Design Review / Critique

Artifacts

- Annotated design files
- Accessibility checklist applied
- Contrast validation references
- Focus order diagrams
- Responsive and zoom behavior notes
- Design accessibility review notes
- Accessibility interaction documentation

Design System Review

Artifacts

- Component accessibility documentation
- Token compliance notes
- Usage guidelines
- Variant and state documentation
- Design component library
- Accessibility annotated components
- Design token contrast validation
- Interaction pattern documentation
- Accessibility behavior specifications for components

Design Hand-Off

Artifacts

- Handoff specification including:
 - Accessible names
 - Roles
 - States
 - Focus behavior
- Keyboard interaction documentation
- Error and validation behavior specification
- Linked Jira tickets
- Accessibility annotations in design files
- Interaction flow diagrams
- Developer implementation notes
- Accessibility acceptance criteria linked to components

Content / Editorial Review

Artifacts

- Plain language review notes
- Heading hierarchy outline
- Alt text draft library

- Readability analysis
- Content accessibility guidelines reference
- Accessible content style guide excerpts

Development Phase

Development Sync / Stand-Up

Artifacts

- Task updates referencing accessibility blockers
- Decision logs for semantic tradeoffs
- Linked implementation resources
- Accessibility issue tracking updates

Technical Design Review

Artifacts

- Engineering design document
- Accessibility implementation plan
- ARIA justification notes
- Focus management strategy
- Interaction event handling documentation
- Accessibility architecture considerations

Code Review / PR Review

Artifacts

- Pull request checklist with accessibility items
- Code comments referencing semantics
- Before and after screen reader test notes
- Linked test results
- Merge request accessibility checklist
- Automated accessibility scan results

- Accessibility linting output
- Code review accessibility feedback notes

Architecture Review

Artifacts

- Framework limitation documentation
- Accessibility constraints analysis
- Dependency risk log
- Accessibility compatibility documentation
- Accessibility infrastructure considerations

Validation Phase

QA / Testing Review

Artifacts

- Accessibility test plan
- Manual keyboard test results
- Screen reader test output
- Defect tickets categorized by severity
- Accessibility test case library
- Assistive technology testing matrix
- Accessibility bug triage notes

Accessibility Review / Audit

Artifacts

- Audit report mapped to WCAG criteria
- Root cause analysis notes
- Recurrence pattern log
- Remediation plan
- Accessibility conformance report draft

- Compliance mapping notes

Regression Testing

Artifacts

- Regression checklist
- Automated scan output
- Baseline comparison log
- Coverage updates
- Accessibility regression test suite
- Accessibility baseline reference results

Deployment Phase

Sprint Review / Demo

Artifacts

- Demo script including accessibility validation
- Feedback capture notes
- Follow-up tickets
- Accessibility behavior validation notes

Release Readiness / Go-No-Go

Artifacts

- Accessibility sign-off checklist
- Known issue register
- Risk acceptance documentation
- Compliance summary
- Accessibility release notes
- Accessibility defect waiver documentation

Continuous Improvement Phase

Retrospective

Artifacts

- Retro notes highlighting accessibility process gaps
- Stage gate improvements
- Action item tracker
- Accessibility improvement backlog

Post-Release Review

Artifacts

- Accessibility metrics dashboard
- Support ticket analysis
- Incident summaries
- Accessibility performance reports

Customer Support / VOC Review

Artifacts

- Tagged accessibility support tickets
- Trend reports
- User feedback summaries
- Accessibility complaint logs

Enablement / Training

Artifacts

- Training materials
- Role-based guidance documentation
- Recorded sessions
- Knowledge check results
- Accessibility onboarding guides

- Accessibility quick-reference guides

Tools

While many tools are used in different capacities, highlight the tools you need to successfully integrate accessibility in your organization or department. Level Access is the number 1 best partner for all your tool needs!

Accessibility Platform Tools

- Ask Level AI
- Evaluations (manual expert testing)
- Scanning & Monitoring
- Level CI
- Platform dashboards and reporting
- Level MCP

Design Tools

- Level Access Figma accessibility plugin
- Figma component libraries
- Figma design tokens
- Figma annotation templates for accessibility requirements
- Adobe XD accessibility plugins
- Sketch accessibility plugins
- Color contrast analyzers for design systems
- Design system documentation platforms

Development Tools

- CI/CD pipeline integrations for accessibility testing
- Code linting tools for accessibility rules
- Git-based merge request / pull request workflows

- Code review templates including accessibility checks
- Design token management systems
- Component libraries with accessibility specifications
- Storybook (or similar) documentation environments
- Frontend accessibility testing frameworks

QA / Testing Tools

- Screen readers (NVDA, JAWS, VoiceOver, TalkBack)
- Keyboard-only navigation testing environments
- Browser developer tools accessibility inspection panels
- Level Access Mobile testing SDK
- Automated accessibility scanning tools integrated into QA workflows
- Accessibility regression testing frameworks
- Assistive technology testing matrices
- Accessibility bug tracking dashboards

Project Management & Collaboration Tools

- Ticket tracking / issue tracking
- DevOps boards
- Repo project boards
- Documentation systems (notion, confluence, etc.)
- Slack / Teams accessibility channels for communications and requests
 - Bonus+ include a ticket system integration for internal review requests
- Shared accessibility issue dashboards

Monitoring & Post-Release Tools

- Production accessibility monitoring systems
- Level Access, automation testing and page scans
- Continuous scanning pipelines

- Error logging systems
- Accessibility metrics dashboards
- User feedback collection systems
- Tracking systems such as Google Analytics or FullStory
- Customer support ticket tagging for accessibility

Documentation & Knowledge Tools

- Accessibility playbook documentation
- Accessibility pattern libraries
- Internal accessibility knowledge bases
- Design system documentation portals
- Accessibility training platforms
- Recorded accessibility training libraries

Research & Validation Tools

- User research platforms
- Usability testing tools
- Session recording tools
- Accessibility participant recruitment tools
- Survey tools capturing assistive technology usage

Reporting & Compliance Tools

- Accessibility conformance reporting tools
- Accessibility issue tracking dashboards
- Compliance documentation systems
- Accessibility scorecards and maturity dashboards

Roles, Personas, and Ownerships

Accessibility succeeds when ownership is clear across teams. The playbook defines which roles are responsible for leading accessibility activities, which roles contribute to them, and where accountability sits at each stage of the lifecycle. Not all roles *should* or *could* exist in one organization so match these to the teams and processes your organization supports. Add some details and

Accessibility Leadership & Governance Roles

- Accessibility Program Lead
- Accessibility Governance Committee
- Accessibility Center of Excellence (CoE)
- Accessibility Champions Network
- Accessibility SME / Subject Matter Expert
- Accessibility Compliance Lead
- Accessibility Legal Advisor
- Accessibility Training & Enablement Lead

Product & Planning Roles

- Product Manager
- Product Owner
- Product Strategist
- Program Manager
- Project Manager
- Scrum Master
- Business Analyst
- Customer Success Representative

Responsibilities often include:

- Defining accessibility requirements
- Ensuring accessibility is included in roadmaps

- Managing risk and prioritization
- Ensuring accessibility acceptance criteria exist

Design Roles

- UX Designer
- Product Designer
- Interaction Designer
- Visual Designer
- Design System Designer
- Content Designer / UX Writer
- Design Researcher
- Accessibility Design Specialist

Responsibilities often include:

- Accessible interaction design
- Design system accessibility
- Component accessibility documentation
- Design annotations for development
- Inclusive research practices

Engineering Roles

- Front-End Developer
- Full Stack Developer
- Design System Engineer
- Accessibility Engineer
- Software Architect
- DevOps Engineer
- Platform Engineer

Responsibilities often include:

- Semantic implementation

- Accessible component development
- Keyboard and focus management
- ARIA usage and validation
- Accessibility automation in CI/CD

QA & Validation Roles

- QA Engineer
- Accessibility Tester
- Automation Test Engineer
- User Acceptance Testing (UAT) Lead
- Accessibility Audit Specialist

Responsibilities often include:

- Manual accessibility testing
- Assistive technology testing
- Regression testing
- Defect tracking and prioritization
- Validation before release

Content & Marketing Roles

- Content Strategist
- Technical Writer
- Marketing Content Lead
- SEO Specialist
- Social Media Manager

Responsibilities often include:

- Accessible written content
- Plain language compliance
- Accessible media and PDFs
- Alt text standards

- Accessible campaign design

Support & Post-Release Roles

- Customer Support Lead
- Customer Support Agent
- Community Manager
- Product Analytics Lead

Responsibilities often include:

- Capturing accessibility feedback
- Tracking accessibility support tickets
- Identifying recurring accessibility issues
- Feeding insights back into the product lifecycle

Executive & Organizational Roles

- Executive Sponsor
- Product Leadership
- Engineering Leadership
- Design Leadership
- Compliance Leadership

Responsibilities often include:

- Setting accessibility expectations
- Allocating resources
- Approving accessibility policies
- Accepting or mitigating accessibility risk

Users & External Personas

Accessibility playbooks must also consider the **users impacted by accessibility decisions**.

- People with visual disabilities
- People with motor disabilities
- People with cognitive disabilities
- People with hearing disabilities
- Assistive technology users
- Temporary impairment users
- Situational impairment users

These personas inform:

- Research
- design decisions
- usability testing
- validation practices