

Instructional Design Tools: Selection and Workflow Design



Select tools for authoring, review, accessibility, and delivery as one toolchain



Treat selection as a systems decision: interoperability, accessibility, security, capacity



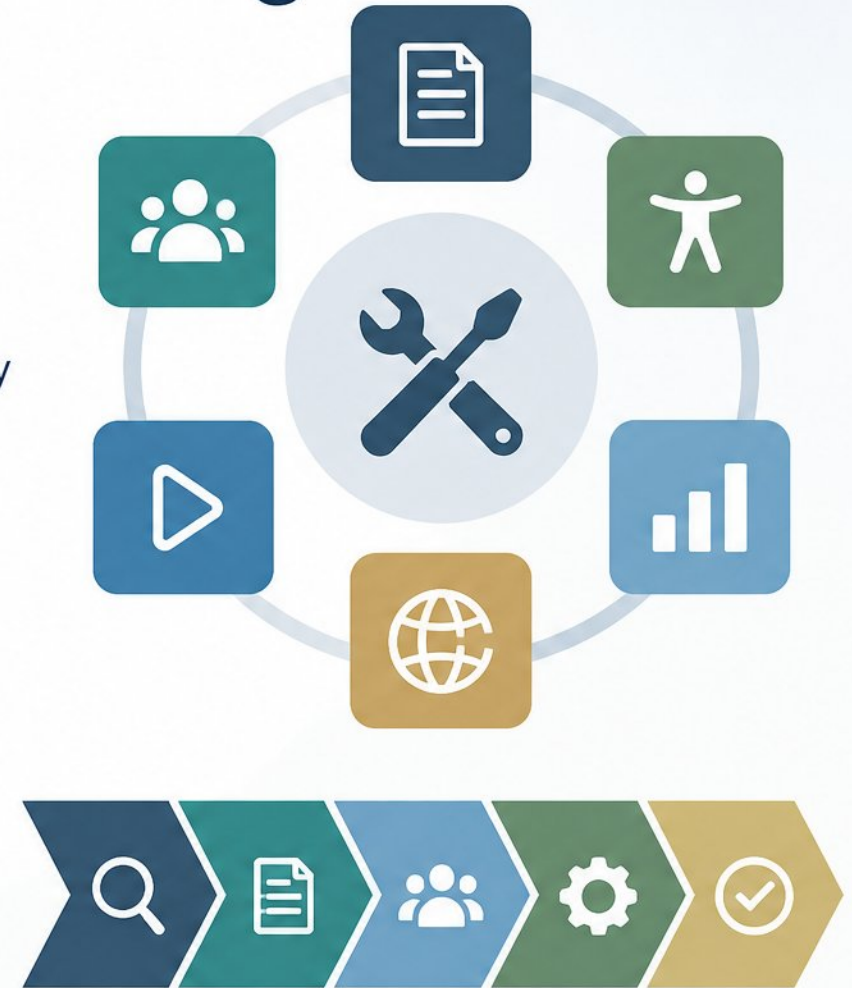
Tool categories: authoring, review, accessibility, LMS/LXP, media, AI drafting, localization, analytics



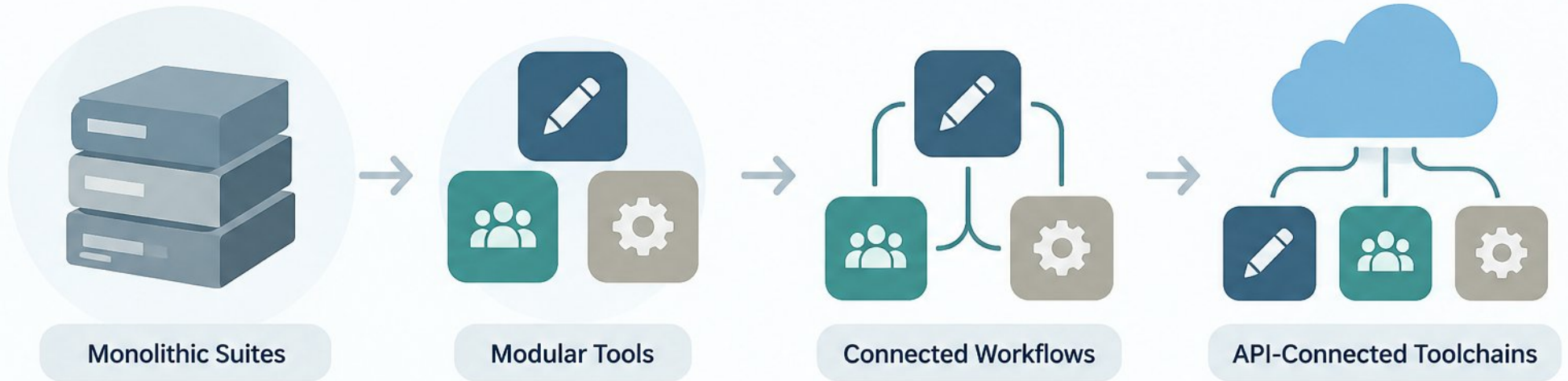
Five-stage roadmap: audit needs, define requirements, evaluate and select, design workflow, operate



Artifacts you will build: requirements matrix, scorecard, workflow map, gate and governance checklists



Background: How the ID Tool Stack Evolved



From monolithic suites to modular, API-connected toolchains



Authoring, review, and accessibility now sit in one flow



AI drafting moved mainstream; accessibility became an in-workflow gate



SCORM stays the LMS baseline; xAPI, cmi5, and LTI drive new selections



Outdated stacks show rework, manual fixes, and reporting gaps

Core Concepts: Tool Categories and Where Each One Fits

Different tools.
A stronger
learning experience
together.



**Authoring
layer**

- Authoring layer: rapid web blocks, desktop freeform, PowerPoint add-ins, browser co-authoring, HTML5 custom builds



**Review
layer**

- Review layer: in-tool annotations, version-controlled document review, prototyping, approval routing



**Accessibility
layer**

- Accessibility layer: built-in checkers, standalone scanners, screen reader and keyboard testing, caption workflows



**Delivery
layer**

- Delivery layer: LMS, LXP, microlearning platforms, virtual classroom, dynamic SCORM packages



**Supporting
layer**

- Supporting layer: media creation, asset libraries, localization, analytics, AI writing and voice



Fit criteria:

- Fit criteria: output fidelity, interaction depth, author skill, integration needs, total cost

Requirements First: Auditing Needs Before Shopping



Start from instructional goals, learner context, and delivery constraints



Functional needs: output standards, interactivity, question types, localization, mobile



Non-functional: security review, data residency, SSO, procurement, total cost of ownership



Accessibility fixed as non-negotiable acceptance criteria up front



Map roles: who authors, reviews, approves, publishes, maintains, owns the contract



Output: weighted requirements matrix, must-have vs. nice-to-have, formally signed off

Weighted Requirements Matrix

	Must-have	Nice-to-have
Functional Needs		
Non-functional		
Accessibility		
Roles		
Other Requirements		



Evaluating and Selecting Tools: A Practical Scorecard



Build a weighted scorecard tied to your requirements matrix



Weight accessibility, interoperability, usability, support, and cost explicitly



Hands-on pilots beat demos: build the same real module in each shortlisted tool



Test completion, scoring, and resume in your actual LMS or LXP



Watch red flags: lock-in, weak ACRs, unclear data ownership, deletion windows



Designing the Workflow: From Intake to Delivery

PEOPLE + PROCESS + CLARITY = BETTER LEARNING OUTCOMES



Map the end-to-end flow: intake, content analysis, storyboard, authoring, review, QA, publish, maintain



Define handoffs, owners, and service-level expectations at each gate



Set version-control and naming conventions; use lean status model: draft to published to archived



Design the review loop deliberately: reviewer roles by round, resolution rules, Definition of Done



Plan integration points across tools; choose metrics before launch: cycle time, revision count, defect rate

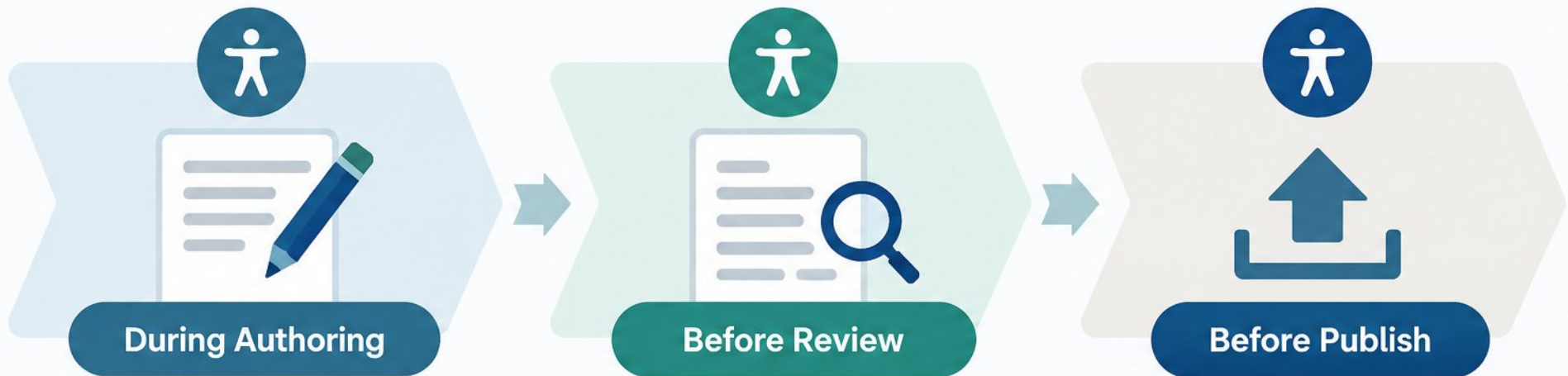


Stage output: a one-page workflow map with swimlanes plus a RACI-style view



A WELL-DESIGNED WORKFLOW TURNS IDEAS INTO IMPACT

Integrating Accessibility Into the Tool Workflow



Embed accessibility checks at three gates: during authoring, before review, before publish.



Verify captions and transcripts; provide alternatives for drag-and-drop and timed assessments.



Automated tools like axe DevTools, WAVE, and Lighthouse catch only 30–40% of issues.



Test the published package in the target LMS — accessibility breaks at launch and on resume.



Manual testing covers the rest: keyboard navigation, screen readers (NVDA, VoiceOver), contrast.



Track defects by severity, document methodology, and keep an updated conformance report.

Selection Implications: Accessibility in Vendor Evaluation



Score accessibility as a weighted procurement criterion, not a checkbox



Require a current ACR/VPAT for the exact product version proposed



Read the ACR critically: template version, date, tester, scope, remarks



Contract specific commitments: WCAG 2.2 AA, defect severity, fix timelines



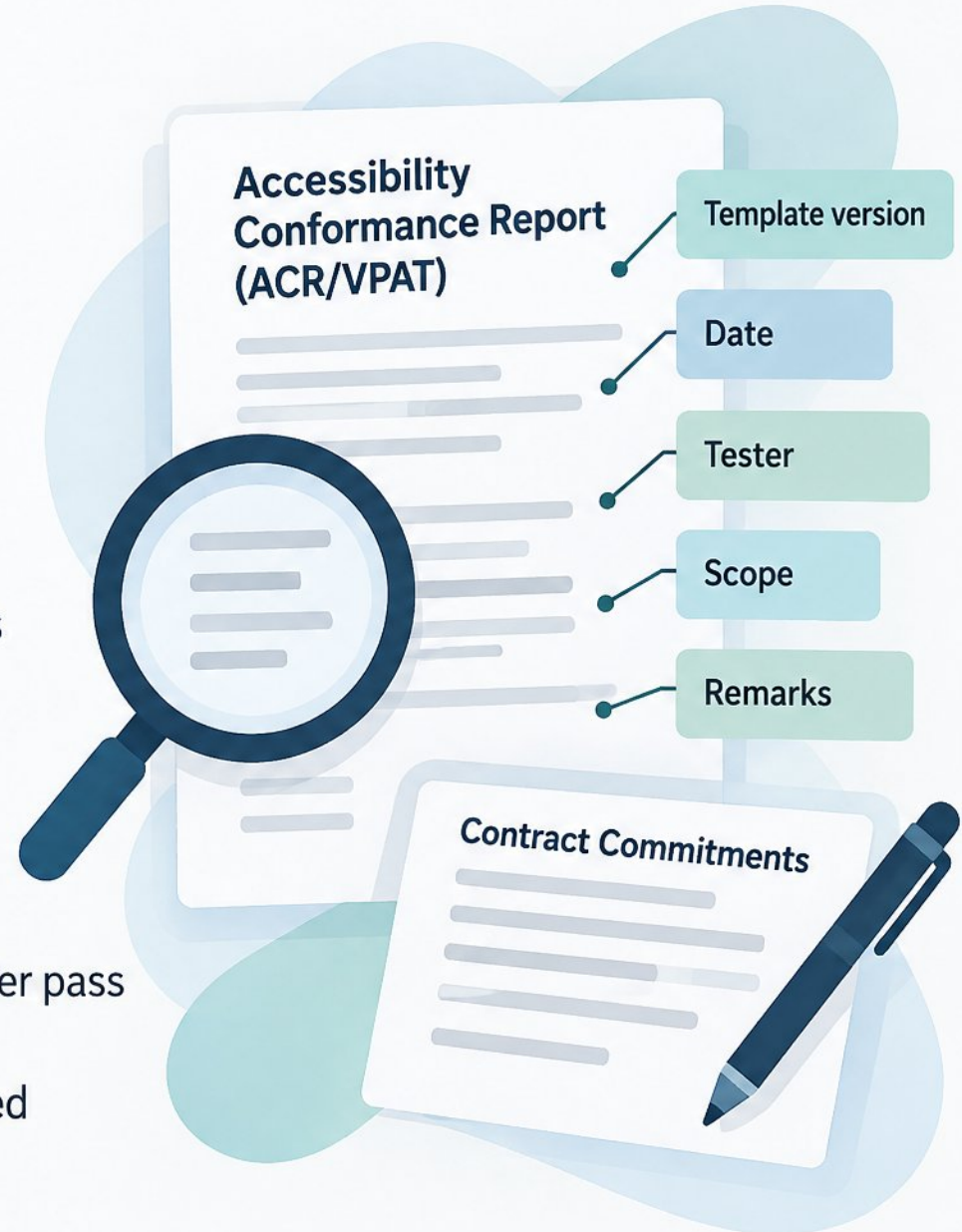
Beware regressions: require no new failures from updates, annual ACR refresh



Test before award: demo environment, automated scan, keyboard and screen reader pass



Prefer platforms with accessibility designed in at authoring time, not retrofitted

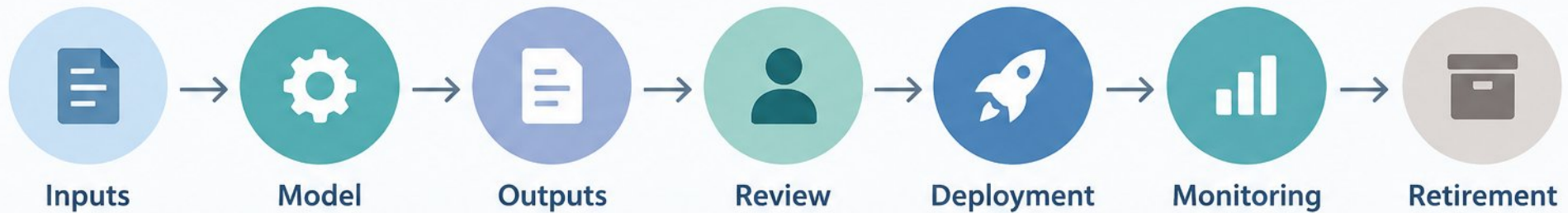


AI-Assisted Authoring: Governance Before Scale

GOVERN THE FULL LIFECYCLE

From idea to retirement, with human oversight.

Responsible
AI for
better learning



AI leads in rapid authoring and localization; humans still own design and facts



Govern the full lifecycle: inputs, model, outputs, review, deployment, monitoring, retirement



Classify by impact: titles low, adaptive paths medium, automated assessment high



Attach proportionate controls: source material, citation checks, expert review, logging



Label AI-assisted content; keep provenance records (model, version, date, oversight role)



Know the legal picture: human authorship, indemnification terms, disclosure duties



Scaling: Governance, Licenses, and Maintenance



Tool governance: named ownership, approved-tool list, exception process, scheduled stack review



Track seat utilization, consolidate overlap, model renewal increases beyond year one



Role-based onboarding, internal templates, workflow docs, office hours for distributed authors



Change management for migrations: communicate early, pilot a real module, keep rollback



Exit planning: data export rights, source-file ownership, and migration paths defined early



Risk, Compliance, and Emerging Tool Considerations



Privacy and security: know how learner data is handled, retained, and where it resides



Standards risk: format drift, export fidelity, and long-term readability of source files



Vendor risk: roadmap shifts, acquisitions, pricing changes, and end-of-life terms



Copyright and licensing: defend what you own in stock, commissioned, and AI-generated assets



Due diligence before approval: security review, conformance evidence, integration test, exit plan



Keep governance live: re-check when laws, vendor terms, or model capabilities change

Common Failure Modes and How to Avoid Them

Common Failure Mode (Before)



Shadow IT



make the approved path
the easy path



Redundant licenses



one inventoried list,
reviewed quarterly



Late accessibility failures



acceptance criteria at
selection and storyboard



Brittle integrations



test the published package
in the real LMS



Unclear ownership



named approvers, fixed review
windows, Definition of Done



Weak enablement



budget training and
documentation as part of adoption

Action Plan: Applying Selection and Workflow Design



First 30 days

- First 30 days: audit current tools, draft requirements matrix, pick one pilot program



Days 30–60

- Days 30–60: build weighted scorecard, test two candidate tools on the same real content



Days 60–90

- Days 60–90: draft one-page workflow map, validate handoffs, define first accessibility gate



- Stand up now: approved-tool list, exception process, named owners, quarterly stack review



- Baseline cycle time, accessibility defect rate, and rework hours; review at 90 days



- Reuse your course artifacts: requirements matrix, scorecard, workflow map with RACI, gate and governance checklists



PersonWise.

PersonWise • AI digital-human course platform



Scan the code or click anywhere to watch this course live, with voice Q&A

personwise.ai/t/95ea42af/public