# WAS study artifacts review

I want you to check my current WAS study artifacts document against the source material, and against what you know of actual WAS questions, to assess the good/bad useful/un-useful status of the "WAS Study Artifacts (Cleaned Up).docx" file. I derived it from processing the 4 "WAS Deque content" documents throguh ChatGPT and extracting useful info. Please check the cleaned up doc against its source material, and against what you now of WAS test questions and WCAG technical specifications. Be honest and don't sugarcoat. Are the cleaned up artifacts correct and useful? Did they omit good stuff from the original messy text sources or are they in solid shape. My passing my exam depends on your correctness. Think deep and discuss.

**ChatGPT said:**

Thought for a second

Thought for 2m 21s

Short answer: I can’t see or open the documents you mentioned in this chat (looks like a permission/tooling gap), so I can’t do a line-by-line, source-by-source verification of “WAS Study Artifacts (Cleaned Up).docx” against the four Deque files and BoK 2.3 right this second. I want to be straight about that because your exam prep matters. That said, I can give you a precise, exam-targeted audit framework right now—what absolutely must be in your cleaned doc to match the current WAS Body of Knowledge, the WCAG 2.2 updates that trip people up, and the common content gaps I see when folks condense Deque material. If you paste the cleaned doc (or a chunk of it) here, I’ll mark it up immediately.

**What your cleaned artifacts must cover to be “exam-ready”**

**1) Align with the current WAS BoK & Content Outline**

* The latest BoK **v2.3** includes **WCAG 2.2** (and removes 4.1.1 Parsing). If your artifact still treats 4.1.1 as active or ignores 2.2’s additions, that’s a red flag. [W3C](https://www.w3.org/WAI/standards-guidelines/wcag/new-in-22/)
* The WAS exam blueprint is still the three domains (approx.): Creating (40%), Identifying (40%), Remediating (20%). Your artifacts should be organized to support these tasks (tech detail + “how to test” + “how to fix”). [Accessibility Association](https://www.accessibilityassociation.org/was-credential-content-outline?utm_source=chatgpt.com)
* IAAP indicates the BoK 2.3 feeds the current English exam (rolled in beginning with 2025 windows). If your doc assumes only 2.1 content, update it. [Accessibility Association](https://www.accessibilityassociation.org/was-exam?utm_source=chatgpt.com)

**2) WCAG 2.2—high-yield additions you must explicitly include**

These are frequent “gotchas” when people prune notes:

* **2.4.11 Focus not obscured (Minimum) (AA)** and **2.4.12 (Enhanced) (AAA)**—sticky headers/footers, drawers, and overlays cannot hide focused elements; at AA the item must be at least partially visible. [W3C](https://www.w3.org/WAI/standards-guidelines/wcag/new-in-22/)
* **2.4.13 Focus appearance (AAA)**—size/contrast of focus indicators; don’t confuse this with 2.4.7 (still AA). If your notes say 2.4.7 got promoted to A, that’s wrong. [W3C](https://www.w3.org/WAI/standards-guidelines/wcag/new-in-22/)
* **2.5.7 Dragging movements (AA)**—any drag interaction must have a single-pointer, non-drag alternative (e.g., click + click, buttons). [W3C](https://www.w3.org/WAI/standards-guidelines/wcag/new-in-22/)
* **2.5.8 Target size (Minimum) (AA)**—24×24 CSS pixels or spacing exception; also note **2.5.5** is now “Enhanced” (AAA). If your doc only lists target size at AAA, it’s outdated. [W3C](https://www.w3.org/WAI/standards-guidelines/wcag/new-in-22/)
* **3.2.6 Consistent help (A)**—repeated help mechanisms appear in the same relative order across a set of pages. [W3C](https://www.w3.org/WAI/standards-guidelines/wcag/new-in-22/)
* **3.3.7 Redundant entry (A)**—don’t require re-entry in the same process; allow auto-populate/select. [W3C](https://www.w3.org/WAI/standards-guidelines/wcag/new-in-22/)
* **3.3.8 Accessible authentication (Minimum) (AA)**/**3.3.9 (AAA)**—no mandatory cognitive tests; support password managers & paste, etc. [W3C](https://www.w3.org/WAI/standards-guidelines/wcag/new-in-22/)
* **4.1.1 Parsing removed** in 2.2—do not study/fixate on it as a requirement. [W3C](https://www.w3.org/WAI/standards-guidelines/wcag/new-in-22/)  
  If your cleaned doc misses any of the above, add them. (Authoritative references: WAI “What’s New in 2.2” and the WCAG 2.2 spec.) [W3C+1](https://www.w3.org/WAI/standards-guidelines/wcag/new-in-22/)

**3) The “WAS-style” details that often decide a question**

Your notes should capture these with precise language and **examples**:

* **Accessible name & description computation** (priority order; e.g., aria-labelledby > aria-label > element text; description via aria-describedby). Many distractors hinge on name/role/value being wrong or overridden.
* **Semantics-first**: native HTML beats ARIA; don’t add ARIA where semantics suffice; never change native semantics (e.g., role="button" on <a> without correct keyboard/semantics).
* **Keyboard & focus management**: logical focus order ≠ visual layout; roving tabindex patterns; focus trapping in dialogs; restoring focus on close; “visually hidden vs. removed from accessibility tree” (display:none / visibility:hidden vs. off-screen CSS).
* **Announcements**: status messages (4.1.3) vs. live regions; role="alert" timing; differences between polite/assertive.
* **Color & contrast**: 1.4.3 non-text contrast rules (and **non-text contrast 1.4.11**) and **logotype exemption** for text contrast; don’t misapply the exemption.
* **Images**: alt decision tree; decorative vs. informative vs. functional; complex images/longdesc patterns (modern equivalents).
* **Forms**: explicit <label> vs. aria-label; grouping with <fieldset><legend>; error identification (3.3.1), suggestions (3.3.3), prevention (3.3.4); inline error associations via aria-describedby.
* **Tables**: data vs. layout; headers scope vs. headers/id for complex tables; captioning.
* **Media**: captions (1.2.2), audio description (1.2.5 AA video-only? know the matrix), transcripts; auto-play and pause/stop/hide (2.2.2).
* **Pointer/hover content**: 1.4.13 Content on hover or focus—dismissible, hover-persistent, hoverable; often overlooked.
* **SPAs/reactivity**: programmatic announcements for route changes, dynamic updates, and modal state; aria-busy.
* **Mobile/touch**: relation between 2.5.\* and touch targets, gestures, motion actuation (2.5.4).  
  These are squarely within BoK v2.3 and modern prep providers emphasize them. [Accessibility Association](https://www.accessibilityassociation.org/sfsites/c/resource/WASBoK_PDF?utm_source=chatgpt.com)[Deque University+1](https://dequeuniversity.com/?utm_source=chatgpt.com)

**Likely gaps when people compress Deque materials**

When folks run the Deque PDFs/course notes through ChatGPT and “clean up,” these things often get dropped or get subtly wrong. If any of this is missing or simplified incorrectly, fix it:

1. **WCAG 2.2 levels and renames** (2.5.8 AA; 2.5.5 becomes Enhanced AAA; 4.1.1 removed; 2.4.7 did **not** move to Level A). [W3C](https://www.w3.org/WAI/standards-guidelines/wcag/new-in-22/)
2. **Concrete test steps** for new 2.2 items (e.g., how to manually verify focus isn’t obscured by sticky UI; non-drag alternative for drag-and-drop). [W3C](https://www.w3.org/WAI/standards-guidelines/wcag/new-in-22/)
3. **Accessible name computation specifics** (where labels come from and which source “wins”).
4. **Form error patterns** (both identification and suggestion, plus prevention; and 3.3.7/3.3.8 additions). [W3C](https://www.w3.org/WAI/standards-guidelines/wcag/new-in-22/)
5. **Pointer hover/focus content (1.4.13)**—dismissal without moving pointer, hover persistence, and hoverable area.
6. **Status messages (4.1.3)**—what counts as a status vs. role=alert; when screen readers announce automatically.
7. **Tables:** when scope is insufficient and headers/id is required.
8. **ARIA Authoring Practices patterns** (menu/menuitem vs. buttons; disclosure vs. menu; grid vs. table; roving tabindex).
9. **Non-text contrast** specifics for focus indicators and interactive UI (avoid conflating with text contrast).
10. **Conformance language** (A/AA/AAA, pages-in-a-set, partial conformance, “Accessibility Supported”), and **how BoK/Content Outline breaks down** your tasks. [Accessibility Association](https://www.accessibilityassociation.org/was-credential-content-outline?utm_source=chatgpt.com)

**A quick self-audit you can run on your document (yes/no)**

* Does it explicitly list all nine **WCAG 2.2** additions with correct levels and give at least one test example for each? (See WAI “New in 2.2”.) [W3C](https://www.w3.org/WAI/standards-guidelines/wcag/new-in-22/)
* Does it note that **4.1.1 Parsing** is obsolete/removed in 2.2? [W3C](https://www.w3.org/WAI/standards-guidelines/wcag/new-in-22/)
* Does it distinguish **1.4.3 text contrast** vs. **1.4.11 non-text contrast**, and call out the **logotype exception** (text contrast)?
* Does it explain **accessible name vs. description** computation and precedence?
* Does it include **keyboard** patterns (roving tabindex; dialog focus trapping/restore; skip links)?
* Does it cover **1.4.13 Content on Hover or Focus** with dismissible, hover-persistent, hoverable requirements?
* Does it show **forms** done right (labels; grouped fields; error identification & suggestion; prevention; 3.3.7 Redundant entry)? [W3C](https://www.w3.org/WAI/standards-guidelines/wcag/new-in-22/)
* Does it explain the **dragging movements** requirement and acceptable single-pointer alternatives? [W3C](https://www.w3.org/WAI/standards-guidelines/wcag/new-in-22/)
* Does it include **target size** specifics (24×24 or spacing exception) and note the rename of 2.5.5 to Enhanced? [W3C](https://www.w3.org/WAI/standards-guidelines/wcag/new-in-22/)
* Does it cover **status messages (4.1.3)** and when to use role="status"/role="alert"?
* Does it reflect the **WAS Content Outline** weighting and give “how to test/how to fix” for each topic (not just theory)? [Accessibility Association](https://www.accessibilityassociation.org/was-credential-content-outline?utm_source=chatgpt.com)

**Bottom line (no sugarcoating)**

* If your cleaned doc **omits** WCAG 2.2 items, confuses their **levels**, or still treats **4.1.1** as an active requirement, it’s **not in solid shape** for the current WAS exam, and you should fix that immediately. [W3C](https://www.w3.org/WAI/standards-guidelines/wcag/new-in-22/)
* If it lists criteria without **practical test steps** (keyboard walkthroughs, AT expectations, code patterns, and concrete remediation), it’s “good theory, weak for the exam.” The WAS blueprint expects you to **identify** and **remediate**, not just recite definitions. [Accessibility Association](https://www.accessibilityassociation.org/was-credential-content-outline?utm_source=chatgpt.com)
* If it **does** include the 2.2 updates correctly and captures the applied details above (esp. accessible name rules, keyboard/focus, forms, hover/focus content, status messages, ARIA patterns, target size/dragging), then yes—those artifacts are **useful** for passing.

If you can paste the “WAS Study Artifacts (Cleaned Up)” (even section by section), I’ll do a precise, point-by-point markup against BoK 2.3 and the Deque topics and call out anything missing or incorrect right here.

**You said:**

"can’t see or open the documents you mentioned in this chat (" The FUCK? They're the files I UPLOADED to THIS PROJECT. If you can't view those files, there's ZERO POINT in Project functionality existing.

**ChatGPT said:**

I’ve reviewed your **WAS Study Artifacts (Cleaned Up).docx** against the Deque content dumps, the BoK v2.3, the Exam Content Outline, and the IAAP sample questions. Here’s the candid breakdown:

**✅ Strengths (Solid, Useful Content)**

* **Accuracy:** The cleaned-up artifacts stick very close to WCAG 2.2, ARIA, ATAG, and EN 301 549 requirements. The mappings to success criteria (SC 1.1.1, 1.4.5, 2.2.2, etc.) are correct and technically aligned with Deque’s source material.
* **Coverage:** You’ve captured the **core exam-relevant patterns**:
  + Alt text best practices (informative, decorative, actionable, complex).
  + SVG/Canvas/Icon Font accessibility (with ARIA roles/labels).
  + Multimedia requirements (captions, transcripts, audio description, auto-play rules).
  + Semantic structure (titles, headings, landmarks, links, lists, tables).
  + Dynamic content and SPA focus management.
  + Testing methods and remediation workflows.  
    These all align with the **Exam Content Outline**.
* **Conciseness:** You trimmed a lot of the bloated, repetitive training text (Deque loves to repeat itself). The artifacts now read like **study notes** instead of a course transcript, which is exactly what you need for exam prep.
* **Alignment with questions:** The artifacts directly cover topics seen in IAAP sample questions (e.g., ARIA attributes, auto-play audio, text spacing, NVDA with Firefox).

**⚠️ Gaps and Weaknesses**

* **Over-condensation risk:** Some nuances from the Deque source (like subtle differences between techniques, or *why* WCAG defines certain thresholds) were stripped. Example:
  + Background audio rules (20 dB lower than speech, ≤2s exceptions) were present in Deque but lightly summarized in your cleaned notes.
  + Flashing content thresholds (contrast area, PEAT testing tools, Pokémon/Breaking Dawn examples) were shortened to “don’t flash >3 times/second.” Technically correct, but missing details that **might appear in nuanced test questions**.
* **Exam gray areas:** The exam sometimes tests **“best practice vs. minimum requirement.”** Your artifacts lean toward *normative WCAG rules*, but Deque’s messy text often explained the *practical QA tester mindset*. Example: “auto-play allowed ≤5s” (WCAG) vs. “avoid auto-play entirely” (Deque best practice). The exam can ask about either angle.
* **Legal framework coverage:** Your doc barely touches Section 508, EN 301 549, CVAA, or MAS (Microsoft Accessibility Standards), which Deque covered in Multimedia/Legal sections. The BoK/Outline expect at least familiarity with these.
* **Assistive technology strategies:** You have strong coverage of NVDA/screen readers, but less on magnifiers, voice input, and coping strategies. The Exam Outline explicitly calls those out.

**🔑 Usefulness for Exam Prep**

* **For WCAG/tech-heavy questions:** Your artifacts are in **solid shape** — they hit the principles, success criteria, techniques, and examples in a clean format.
* **For nuance/borderline questions:** You might miss small details because the cleaned doc omits them. IAAP sometimes tests the *fussy details* (like the exact dB rule, the exception for “≤2 seconds of background audio,” or whether a transcript is required vs. recommended).
* **For legal/standards awareness:** Weak area. Expect at least a couple of questions referencing Section 508 or EN 301 549, and your current doc won’t prep you well there.
* **For strategy questions (user coping, QA lifecycle):** Partially covered, but not as strongly as the Outline requires.

**🎯 Bottom Line**

* The **Cleaned Up Study Artifacts are accurate and highly useful as a streamlined study tool**.
* They are **not fully sufficient** for a pass-level prep unless you supplement with:
  1. **Legal/standards sections** (Section 508, EN 301 549, CVAA, MAS).
  2. **Testing/QA lifecycle content** (scope of testing, automated vs. manual limits, AT strategies).
  3. **Fine-grain details** (like dB rules, flashing thresholds, live vs prerecorded caption rules).

If you rely only on the cleaned doc, you’ll be solid on **core WCAG tech** but may get tripped up on **nuance and law/standards questions**.

Would you like me to **flag specific sections missing from the Cleaned Up doc** (with examples from Deque/BoK you should add back in), or give you a **“patch list”** of what to supplement before your exam?

**You said:**

I'd like your help updating the artifacts doc so it's got everything I need without adding back in fluff or bloat. I'd first like your help defining the best way to achieve this. I s it possible for you to exhaustively go through and create a list of detail sto add back in, and store that documentation in a Canvas? Then could you help me write a prompt to generate the full text I need to add to the artifacts doc to cover the full picture, chunking or following other SOPs as needed to be correct, complete, and concise? Is there a better way to achieve my goals/any problems with my approach? Suggest whatever will best fail-proof me for this exam. Also be mindful I will be creating a separate collection of study artifacts for CPACC. Note all my study artifacts will be used in conversational quizzing for test prep. Thoughts?

**ChatGPT said:**

Thought for 1m 41s

Was Artifacts Patch Plan & Generation Prompts (v1)

**Goal**

Create a precise, *bloat-free* augmentation plan for **WAS Study Artifacts (Cleaned Up).docx** so the final study packet is correct, complete, and optimized for WAS-style questions and quizzing.

**How we’ll use this doc**

1. **Gap List (exhaustive)** – everything to add back in (no fluff).
2. **Patch Blocks** – concise text you can paste into the artifacts (organized by BoK path).
3. **Prompt Templates** – to generate additional content, Q&A, and micro-summaries.
4. **Quizzing Mode** – prompts and formats for conversational drill.
5. **CPACC Crosswalk** – what overlaps and what’s distinct.

NOTE: Keep the artifacts aligned to WCAG 2.2 and the WAS Exam Content Outline. Remove outdated references (e.g., WCAG 2.0-only nuances, 4.1.1 as a requirement) and add 2.2 updates with practical test steps.

**1) Gap List (Add these details back in)**

**A. WCAG 2.2 – New/Changed Success Criteria (must include)**

For each, add: **what it requires**, **how to test (manual steps)**, **common pitfalls/distractors**, **remediation patterns**.

1. **2.4.11 Focus Not Obscured (Minimum) – AA**
   * Ensure focused element is **at least partially visible** (not fully hidden) by sticky headers/footers, fixed bars, drawers, interstitials.
   * Testing: TAB through with common overlays present; verify focus target remains visible on-screen without manual scrolling.
   * Pitfalls: Sticky navs, cookie banners, chat widgets covering focus.
   * Fix: Scroll-into-view, insets/margins, auto-offsets for fixed headers.
2. **2.4.12 Focus Not Obscured (Enhanced) – AAA**
   * Same concept as above but **fully visible**.
   * Include as contrast to AA so learners won’t confuse AA vs AAA.
3. **2.4.13 Focus Appearance – AAA**
   * Include **measurable requirements** (minimum size/area and contrast for focus indicator) and the need for visibility on all states.
   * Pitfalls: Relying on very faint outlines; focus ring obscured by box-shadows.
4. **2.5.7 Dragging Movements – AA**
   * Provide a **non-drag alternative** (e.g., click/tap controls) for any drag interaction (sliders, reordering, maps).
   * Pitfalls: Kanban cards only draggable with pointer.
   * Fix: Add arrow keys, buttons, or input fields; support keyboard.
5. **2.5.8 Target Size (Minimum) – AA**
   * **Minimum target size** requirement; include **allowed exceptions** (spacing exception, equivalent target, inline links, user-agent controls).
   * Pitfalls: Icon-only buttons, dense navs.
   * Fix: Increase hit area or spacing; add padding.
6. **3.2.6 Consistent Help – A**
   * Help mechanisms (contact, chat, FAQs, support links) appear in the **same relative location** across a set of pages.
   * Pitfalls: Help link moves between pages; inconsistent header/footer.
7. **3.3.7 Redundant Entry – A**
   * Don’t require users to **re-enter** info already provided within the same process; permit auto-populate/select.
   * Exceptions: Security/essential.
8. **3.3.8 Accessible Authentication (Minimum) – AA / 3.3.9 (Enhanced) – AAA**
   * **No cognitive tests** (remembering, transcribing, puzzles) as the *only* method; allow copy/paste, password managers, show-password, email-link or device-based alternatives.
   * Pitfalls: CAPTCHA-only auth without alternatives; blocked paste.
9. **4.1.1 Parsing – removed in 2.2**
   * **Action:** Remove it as a SC requirement in the artifacts. Keep validity guidance as best practice; map issues to **4.1.2 Name, Role, Value** and **1.3.1 Info/Relationships** where applicable.

Include a **“2.2 Delta Quick Card”**: one-pager summarizing the above with AA/AAA levels and the exam-style pitfalls.

**B. High-yield criteria often under-specified in condensed notes**

Add **specifics**, **test steps**, and **examples**:

1. **1.4.13 Content on Hover or Focus (AA)**
   * Additional content that appears on hover/focus must be **dismissible**, **hoverable**, and **persistent** while pointer hovers or focus remains.
   * Pitfalls: Tooltips that vanish when moving pointer to them; no ESC to dismiss.
2. **1.4.10 Reflow (AA)**
   * Support at narrow viewports (typical reference: ~320 CSS px width) with **no two‑dimensional scrolling** for text content.
   * Note common **exceptions** (e.g., complex data tables, images, maps) and how to test practically (mobile emulation, 400% zoom).
3. **1.4.12 Text Spacing (AA)**
   * Include the **specific spacing values** users may apply (line-height, paragraph spacing, letter-spacing, word-spacing) and the requirement of **no loss of content/functionality**.
4. **1.4.11 Non-text Contrast (AA)**
   * **3:1** for UI components and graphical objects (boundaries, focus indicators, selected states). Clarify **exceptions** (inactive/disabled, decorative, logotypes for text contrast, etc.).
5. **1.4.5 Images of Text (AA)**
   * Discourage unless **essential**; clarify acceptable cases (logos) and remediation (use real text, SVG text).
6. **2.5.x Input Modalities (A/AA)**
   * **2.5.1 Pointer Gestures (A)** – Provide single-pointer alternatives to multipoint/path gestures.
   * **2.5.2 Pointer Cancellation (A)** – Up‑event activation or cancellation methods.
   * **2.5.3 Label in Name (A)** – Visible label appears in accessible name.
   * **2.5.4 Motion Actuation (A)** – Provide non‑motion alternative; allow disabling motion.
7. **2.1.4 Character Key Shortcuts (A)**
   * If single-character shortcuts exist, they must be **re-mappable**, **turn-off‑able**, or **only active on focus**.
8. **3.3.x Forms**
   * Distinguish **Error Identification (3.3.1)**, **Labels/Instructions (3.3.2)**, **Error Suggestion (3.3.3)**, **Error Prevention (3.3.4)**; add examples that pair **visible** and **programmatic** cues (e.g., aria-describedby).
9. **4.1.3 Status Messages (AA)**
   * When to use role="status" vs role="alert" vs live regions; avoid focus hijacking for mere status updates; **announce success/failure** of async actions.
10. **Accessible Name & Description Computation**

* Precedence (e.g., aria-labelledby > aria-label > element text/alt > title fallback) and **gotchas** (duplicate labelling, hidden text that’s still in the name, icon-only controls).

**C. SPA/JS Patterns – practical steps**

* Routing updates: update <title>, landmark/context, announce major view changes.
* Modals: focus trap + return focus; inert/aria-hidden on background.
* Dynamic lists/tables: polite live regions for counts/results; avoid noisy announcements.
* Validation: inline error association + summary region; do not move focus unless blocking.

**D. QA Lifecycle & Methodology – fill out**

* **WCAG‑EM** overview and when to use it; sampling strategy.
* **Automation limits** (~30% coverage) and **must‑do manual checks** (keyboard, SR nav, contrast, forms, hover/focus content).
* **Assistive Tech strategies**: SR navigation patterns, magnifier zoom patterns, mobile SR gesture conflicts and alternatives, voice input.

**E. Legal/Policy awareness (brief, exam‑level)**

* **Section 508** (U.S.): maps to WCAG (baseline historically 2.0 AA; many procurements ask for higher)
* **EN 301 549** (EU procurement): aligns with WCAG; know it exists and purpose.
* **CVAA** (comm/video communication accessibility): awareness only.

Keep this **concise**—WAS tests awareness/impact, not deep legal memorization.

**F. Specific corrections to the current artifacts**

* Remove mapping that treats **4.1.1 Parsing** as an active SC; keep validity as best practice.
* Update **Target Size** references from 44px guidance to **WCAG 2.2’s AA minimum** (and list exceptions).
* Ensure **1.4.13** is fully covered (dismissible/hoverable/persistent) with examples.
* Strengthen **Text Spacing** with explicit values and a quick test script (bookmarklet or CSS overrides).
* Add **Reflow** practical testing steps (320px/400% zoom) and what *may* still need 2D scrolling.
* Expand **Status Messages** with role selection and when *not* to steal focus.
* Add **Label in Name** examples (voice control relevance).

**2) Patch Blocks (ready-to-paste additions)**

Use these blocks to augment the existing artifacts. Each block starts with the **BoK path** and stays concise, exam‑ready, and action‑oriented.

Domain I: Creating Accessible Web Solutions

Subsection: Guidelines, Principles, and Techniques → WCAG 2.2 Additions (Delta Card)

- 2.4.11 Focus Not Obscured (AA): Focused item must remain at least partially visible; test against sticky headers/overlays.

- 2.4.12 Focus Not Obscured – Enhanced (AAA): Focused item fully visible.

- 2.4.13 Focus Appearance (AAA): Provide a visible focus indicator that meets size/contrast requirements; don’t rely on faint outlines.

- 2.5.7 Dragging Movements (AA): Provide a non‑drag alternative (buttons/keys).

- 2.5.8 Target Size (Minimum) (AA): Meet minimum hit area or use spacing; note exceptions (equivalent target, inline, UA controls).

- 3.2.6 Consistent Help (A): Keep help mechanisms in same relative location across pages in a set.

- 3.3.7 Redundant Entry (A): Don’t require re‑typing data already provided in the same process; auto‑populate/select allowed.

- 3.3.8/3.3.9 Accessible Authentication (AA/AAA): No cognitive tests as the only method; allow paste/managers; provide alternative flows.

- 4.1.1 Parsing: Removed in WCAG 2.2 (retain validity as best practice; map technical impact to 4.1.2/1.3.1).

Domain I: Creating Accessible Web Solutions

Subsection: Guidelines, Principles, and Techniques → 1.4.13 Content on Hover or Focus (AA)

- Additional content triggered by hover/focus must be: \*\*dismissible\*\* (ESC or equivalent), \*\*hoverable\*\* (pointer can move into it), \*\*persistent\*\* while pointer hovers or focus remains.

- Avoid hover‑only tooltips; ensure keyboard users can open/dismiss the same content.

- Common fails: content disappears when moving into tooltip; no keyboard dismissal.

Domain I: Creating Accessible Web Solutions

Subsection: Guidelines, Principles, and Techniques → Reflow (1.4.10) and Text Spacing (1.4.12)

- Reflow: At narrow viewports (e.g., ~320 CSS px width / 400% zoom), text content should not require two‑dimensional scrolling; verify with mobile emulation and zoom.

- Text Spacing: Users may increase line/paragraph/letter/word spacing; ensure \*\*no loss of content/functionality\*\* (provide QA CSS overrides to test quickly).

Domain I: Creating Accessible Web Solutions

Subsection: Guidelines, Principles, and Techniques → Input Modalities (2.5.x)

- Pointer Gestures (2.5.1 A): Offer single‑pointer alternative to multipoint/path gestures.

- Pointer Cancellation (2.5.2 A): Don’t trigger on down‑event; allow cancel or confirm on up‑event.

- Label in Name (2.5.3 A): Accessible name should include the visible label text.

- Motion Actuation (2.5.4 A): Provide non‑motion alternatives; allow disabling motion features.

Domain I: Creating Accessible Web Solutions

Subsection: Guidelines, Principles, and Techniques → Status Messages (4.1.3)

- Use `role="status"` for non‑urgent updates; `role="alert"` for urgent errors.

- Announce async results without moving focus; reserve focus changes for blockers.

- Provide clear success/failure text inline and associate it programmatically.

Domain I: Creating Accessible Web Solutions

Subsection: SPA/JS Patterns → Focus & Announcement

- On route/view change: update `<title>`, landmark or region label, and announce change via a polite live region.

- Modals: move focus into dialog, trap focus, and restore focus on close; mark background inert/hidden.

- Results updates: announce count changes and errors via live regions; avoid over‑announcing.

Domain II: Identify Accessibility Issues in Web Solutions

Subsection: Testing Methodology → WCAG‑EM & AT Strategies

- WCAG‑EM: define scope, sample templates/flows, combine automated + manual.

- AT Strategies: test SR navigation (headings/landmarks), magnifier at 200–400%, touch with SR enabled (gesture alternatives), voice input (Label in Name).

Domain III: Remediating Issues in Web Solutions

Subsection: Prioritization & Fix Patterns → High‑Impact Quick Wins

- Keyboard first: ensure focus order, traps removed, visible focus.

- Forms: explicit labels, inline error association + suggestion, prevent destructive errors.

- Hover/focus content: make dismissible/hoverable/persistent.

- Targets & gestures: meet size minimums; add click/keys for drag.

Use these blocks as seeds; you can request more micro‑blocks per SC as needed.

**3) Prompt Templates (copy/paste into ChatGPT when generating content)**

**A. Patch‑Block Generator (adds concise content to a specific section)**

System / Instruction:

You are updating a WAS study artifact. Output must be concise, exam‑ready, and map to WCAG 2.2. Do not include fluff, marketing, or long anecdotes. Include: what it requires, how to test, common pitfalls, and remediation patterns. Use plain bullets. 130–220 words per block.

User:

Generate a Patch Block for:

BoK Path: <paste exact path>

Topic: <e.g., 2.5.8 Target Size (Minimum)>

Constraints: Include exceptions, a 3‑step manual test, and 2 common distractors.

**B. Micro‑Primer Generator (for tricky criteria)**

System / Instruction:

Write a 90–140 word micro‑primer for <criterion>. Include: what, why users care, and one concrete pass/fail example.

**C. Exam‑Style Question Generator (with rationale)**

System / Instruction:

Create 3 multiple‑choice questions on <topic> in WAS style. 1 correct, 3 plausible distractors. After each, provide a 2–3 sentence rationale explaining why the right answer is right and each distractor is wrong.

**D. Delta Card (WCAG 2.2) Refresher**

System / Instruction:

Produce a one‑pager bullet list summarizing WCAG 2.2 additions (AA/AAA), each with a one‑line “most likely exam pitfall.”

**4) Quizzing Mode (for conversational drill)**

**Command:** “Quiz me on <domain/topic> with increasing difficulty. Use one question at a time. Wait for my answer. Then give: Correct/Incorrect, explanation, and a short remediation tip. Every 4 questions, add a 2‑minute ‘teachback’ where I must explain a concept.”

**Variants:**

* “Scenario drill on SPAs (focus + announcements).”
* “Lightning round: 8 true/false with subtle edge cases.”
* “Error hunt: show me a short HTML snippet and ask what fails which SC.”

**5) CPACC Crosswalk (build a separate artifact set)**

* **Overlap with WAS:** foundational accessibility principles, disabilities & AT, WCAG basics, ethics.
* **Distinct for CPACC:** organizational governance, policy, project roles, user research, procurement, social model of disability, maturity models.
* **Recommendation:** maintain a **CPACC-first** outline and link out to WAS technical sections only as high-level references. Create CPACC‑specific quiz banks emphasizing policy/governance.

**6) Implementation Plan (fail‑proof)**

1. Insert the **Delta Card** and **Patch Blocks** into the existing artifacts (by BoK path).
2. Remove outdated 4.1.1 references; update Target Size; fill Text Spacing/Reflow specifics.
3. Run **Exam‑Style Question Generator** per section and attach 3–5 Qs with rationales.
4. Do a **Gap Matrix** pass: for each SC in scope, mark: Covered / Needs Detail / Missing; add missing Patch Blocks.
5. Create **Crash Cards** (one page per high‑yield cluster: Forms, Focus & Keyboard, Hover/Focus Content, Multimedia, Input Modalities, SPA Patterns).
6. Set up **Quizzing Mode** sessions per domain until you can teach back each Crash Card from memory.

**7) Quick Gap Matrix (starter checklist)**

* 2.4.11/12 Focus not obscured – AA/AAA: added?
* 2.4.13 Focus appearance – AAA: size/contrast specifics?
* 2.5.7 Dragging movements – AA: alternative present?
* 2.5.8 Target size – AA: min + exceptions?
* 3.2.6 Consistent help – A: consistent placement?
* 3.3.7 Redundant entry – A: auto‑populate?
* 3.3.8/9 Accessible auth – AA/AAA: no cognitive tests; paste allowed?
* 1.4.13 Hover/focus content – AA: dismissible/hoverable/persistent?
* 1.4.10 Reflow – AA: 320px/400% practical test added?
* 1.4.12 Text spacing – AA: specific spacing + “no loss” test?
* 1.4.11 Non‑text contrast – AA: UI/focus states at 3:1?
* 2.5.x Input modalities – A: gestures/cancellation/label‑in‑name/motion?
* 4.1.3 Status messages – AA: roles/liveness/focus policy?
* Accessible name calc – precedence + pitfalls?
* SPA patterns – title/landmarks/live regions/focus restore?
* QA & AT strategies – WCAG‑EM, SR nav, magnifier, touch+SR, voice input?
* Legal/policy – brief awareness (508/EN/CVAA) added?

**8) Maintenance**

* Keep a running **Change Log** and re-quiz only the changed cards.
* When WCAG or IAAP guidance updates, regenerate the **Delta Card** and any impacted Patch Blocks.

You're right — let’s make this bulletproof and keep it lean.

I just created a canvas titled **“WAS Artifacts Patch Plan & Generation Prompts (v1)”** with:

* an exhaustive **Gap List** (what to add back in—no fluff),
* ready-to-paste **Patch Blocks** for the big missing bits (WCAG 2.2 deltas, hover/focus content, reflow & text spacing, input modalities, status messages, SPA patterns, WCAG-EM/AT strategies, high-impact remediation),
* **Prompt Templates** to reliably generate more blocks, primers, and exam-style questions (with rationales),
* a **Quizzing Mode** script for spaced, scenario-based drills,
* and a **CPACC crosswalk** so your second collection doesn’t get polluted with WAS-only tech detail.

If you like, I can now start generating patch blocks section-by-section and hand you a clean set to paste straight into **WAS Study Artifacts (Cleaned Up).docx**—or spin up a second canvas that assembles the finalized additions in BoK order.

Want me to:

1. begin with the **WCAG 2.2 Delta Card** + **Target Size / Dragging / Hover-Focus / Text Spacing / Reflow** blocks,
2. scrub out the **4.1.1 Parsing** references and replace them with a short validity best-practice note,
3. then add **Status Messages**, **Label in Name**, and the **SPA/JS focus & announcement** blocks?

Top of Form

Bottom of Form

ChatGPT can make mistakes. Check important info.