# 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 Remove outdated references (e.g., WCAG 2.0-only nuances, 4.1.1 as a requirement) and add 2.2 updates with practical test steps.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.