**0) Operating Model**

**Cadence:** weekly sprints (adjust to 1–2 weeks if needed).

**Ceremonies:**

* Sprint Planning (90m), Daily (15m), Mid-sprint Review (30m), Sprint Review/Demo (45m), Retro (30m).

**Boards:** Product (Epics), Sprint (Stories/Tasks).

**Branching:** main (prod), short-lived feature branches, PR <300 LOC, one concern.

**DoR (Definition of Ready):** user value + AC (Given/When/Then) + test notes + no external blockers.

**DoD (Definition of Done):** code + tests + docs + Actions green + **SHA proof** (/api/version == HEAD SHA) + **UAT Gate** passed in staging & prod-safe + demo recorded.

**Constraints (must always apply):**

* Re-read **README**, /docs, and **/DEVELOPMENT\_GUIDELINES.md** before **every** change.
* **No hacks/workarounds**. Fix **root causes** only.
* **Actions** is canonical; post-deploy **UAT Gate** is mandatory.

**1) Epics (Vision-aligned)**

**E1. CI/CD & Environment Baseline (Deterministic Releases)**

*Deployments are reliable, SHA-proven; dual infra drift removed.*

**Success:** Production workflow builds web image tag=${{ github.sha }}, wires, sets build SHA, and proves /api/version == sha.

**E2. Auth Lifecycle Hardening (AAD-Only Prod)**

*Sign-in/out deterministic; no demo in prod; no 0.0.0.0 callbacks.*

**Success:** AAD round-trip & sign-out are reliable; no auto-login without Microsoft round-trip.

**E3. API Health & Contracts (Presets/Assessments/Engagements/Admin)**

*UI↔API contracts via /api/proxy are consistent and resilient.*

**Success:** Presets list/get/upload 2xx; create→get 201→200; engagements list 2xx; /api/proxy/admin/status 200 for Admin.

**E4. Chat Orchestrator v1 (Conversation & Tool Routing)**

*Minimal “brain” to route intents to tools; maintain conversation state.*

**Success:** Chat can call Doc Analyzer & Transcribe→Minutes tools and persist run logs.

**E5. Evidence Ingestion & Mapping v1**

*Ingest docs; map to maturity controls; evidence traceability.*

**Success:** Uploaded docs produce control coverage with citations.

**E6. Maturity Scoring v1**

*Score domains from mapped evidence; explainability with gaps.*

**Success:** Domain levels computed with “why” + missing evidence list.

**E7. Roadmap & Exports v1 (PPTX/Jira)**

*From gaps to prioritized roadmap with cost/effort & exports.*

**Success:** PPTX deck + Jira epics/issues generated from roadmap.

**E8. Engagement RAG & Citations**

*Per-engagement index used by chat/Doc Analyzer for citations.*

**Success:** Answers contain citations to evidence; contradiction prompts.

**E9. Admin & Presets Management**

*Admin can curate models/presets; imports validated & versioned.*

**Success:** Admin-presets CRUD & validation; canonical presets seeded.

**E10. Consent & GDPR**

*Consent workflow for audio/docs; privacy TTL/audit dashboards.*

**Success:** Consent stored & auditable; GDPR Ops dashboards render.

**E11. Observability & SLOs**

*Structured logs, correlation IDs, SLO dashboards; alerts.*

**Success:** Error budgets & alerts configured; support bundle script.

**E12. Security & Compliance**

*CRITICAL/HIGH security backlog closed; dependency/license guardrails.*

**Success:** No high/critical open; scans & policy checks green.

**E13. Repo Hygiene & Docs**

*Size guards; dead docs removed; single source-of-truth docs; PR checklist.*

**Success:** Repo lean; docs current; PR template enforced.

**2) Release Plan (10 Sprints, weekly; each ends with deploy + UAT + demo)**

Story points: use **S**=1, **M**=3, **L**=5 (rough). All stories include tests & docs.

“Artifacts” = attach in PR: test results, Actions run, UAT screenshots/videos, updated docs.

**Sprint 1 — Stabilize Releases & Sign-in/out Baseline**

*Epics:* E1, E2 (foundation)

* **S1-1 (E1, M):** Production workflow **must** set build SHA and assert GET /api/version == sha.

**AC:** Failing proof fails job; passing proof marks green.

**Artifacts:** run log, proof screenshot.

* **S1-2 (E2, S):** /api/auth/providers shows **azure-ad** only in prod; no demo.

**AC:** prod JSON shows { "azure-ad": … } only; PR links to output.

* **S1-3 (E2, M):** Sign-out clears session; lands on /signin at Container Apps host.

**AC:** No redirect to 0.0.0.0/localhost; re-clicking AAD requires Microsoft round-trip.

**Artifacts:** Playwright auth smoke video.

* **S1-4 (E1, S):** Remove **App Service API** references from workflows/docs/scripts.

**AC:** grep shows none; docs updated.

**UAT:** AAD sign-in/out; evidence no regressions.

**Demo:** “Reliable deploy + real sign-out.”

**Sprint 2 — Contracts: Presets/Assessments/Engagements Admin Status**

*Epics:* E3

* **S2-1 (E3, M):** /api/proxy/presets list/get return 2xx; seed includes cscm-v3, cyber-for-ai.

**AC:** UI table non-empty; GET id works.

* **S2-2 (E3, M):** Preset upload returns 201 and appears in list.

**AC:** Upload 201; retrievable by id.

* **S2-3 (E3, M):** Assessment **create→get** 201→200; ID shape consistent (engagement-…).

**AC:** UI detail page loads without “Failed to get assessment.”

* **S2-4 (E3, S):** /api/proxy/admin/status 200 for Admin session (server-verified).

**AC:** Playwright smoke: admin Ops/GDPR no longer “Access denied.”

**UAT:** Create assessment and open it; admin pages render.

**Demo:** “Preset catalog & create→read assessment.”

**Sprint 3 — Chat Orchestrator v1 (Conversation + Minimal Routing)**

*Epics:* E4

* **S3-1 (E4, M):** Conversation store per engagement (metadata, participants, run-logs).

**AC:** Inspectable JSON; persisted.

* **S3-2 (E4, M):** Intent→tool routing minimal: “analyze doc” + “transcribe minutes”.

**AC:** Chat commands invoke tools; run logs captured.

* **S3-3 (E4, S):** Guardrails: RBAC + consent flag check per tool.

**AC:** Unauthorized tool calls blocked with clear message.

**UAT:** Chat invokes Doc Analyzer with a doc; Transcribe with a short audio; logs visible.

**Demo:** “Hello orchestrator.”

**Sprint 4 — Evidence Mapping v1**

*Epics:* E5

* **S4-1 (E5, L):** Doc Analyzer maps controls for active model; store traceability (citations).

**AC:** Control coverage list with evidence snippet refs.

* **S4-2 (E5, M):** Evidence overview page under engagement (tab).

**AC:** List shows documents, mapped controls, confidence.

**UAT:** Upload doc, chat analyze → mapped controls visible with citations.

**Demo:** “Evidence to controls.”

**Sprint 5 — Maturity Scoring v1**

*Epics:* E6

* **S5-1 (E6, L):** Scoring agent computes domain levels with rationale & missing-evidence prompts.

**AC:** JSON result persisted; UI renders domain radar/table; reasons visible.

* **S5-2 (E6, M):** “Ask for what’s missing” prompts added to chat.

**AC:** Chat returns 2–3 targeted follow-ups.

**UAT:** From mapped evidence, score maturity; see gaps & prompts.

**Demo:** “Your current maturity, here’s why.”

**Sprint 6 — Roadmap & Exports v1 (PPTX/Jira)**

*Epics:* E7

* **S6-1 (E7, L):** Roadmap generator with cost bands & resource profiles.

**AC:** Prioritized, impact/effort, dependencies, ETA.

* **S6-2 (E7, M):** Export: PPTX deck (title, scores, roadmap).

**AC:** Downloadable PPTX with correct sections.

* **S6-3 (E7, M):** Export to Jira: epics/issues with idempotency.

**AC:** Jira issues created once; re-runs update, not duplicate.

**UAT:** Generate roadmap from current scores; export PPTX & Jira.

**Demo:** “From gaps to an exec deck & tickets.”

**Sprint 7 — RAG Index & Citations**

*Epics:* E8

* **S7-1 (E8, L):** Per-engagement RAG index; chat & Doc Analyzer cite sources.

**AC:** Answers show citations; clicking opens snippet.

* **S7-2 (E8, M):** Contradictions agent highlights conflicts across evidence/minutes.

**AC:** Contradiction list rendered; follow-up questions in chat.

**UAT:** Ask “why level X?” → cited answer; contradictions surfaced.

**Demo:** “Show me the evidence.”

**Sprint 8 — Admin & Presets Management**

*Epics:* E9

* **S8-1 (E9, M):** Admin presets CRUD (validate, version, owner, notes).

**AC:** Create/update works; invalid JSON rejected with messages.

* **S8-2 (E9, S):** Seed canonical presets in prod (idempotent).

**AC:** Seeds appear once; re-run safe.

**UAT:** Admin imports/edits preset; appears in user flow.

**Demo:** “Curate your maturity model presets.”

**Sprint 9 — Consent & GDPR**

*Epics:* E10

* **S9-1 (E10, M):** Consent UI: capture, store, link to artifacts (audio/docs).

**AC:** Audit entry created & visible; PII scrub step enforced.

* **S9-2 (E10, M):** GDPR Ops dashboards render (TTL, audit logs, jobs).

**AC:** Admin pages no longer “Access denied”; show data.

**UAT:** Ingest audio with consent; verify GDPR page shows record.

**Demo:** “Privacy built in.”

**Sprint 10 — Observability, Security, Hygiene**

*Epics:* E11, E12, E13

* **S10-1 (E11, M):** Add SLO dashboards & alerts; support bundle script.

**AC:** Alert triggers on error budget breach; bundle zips logs/config safely.

* **S10-2 (E12, M):** Close remaining HIGH security findings (injection/path/CORS).

**AC:** Security scans green; zero high/critical.

* **S10-3 (E13, S):** CI size guard + .gitignore audit; stalebot for PRs/branches.

**AC:** Large files blocked; branch/PR hygiene auto-managed.

**UAT:** Non-functional: alarms tested on staging; size guard blocks test artifact commit.

**Demo:** “We can operate & keep it healthy.”

**3) Backlog by Epic (User Stories, Acceptance Criteria, Dependencies)**

Below are representative stories (you can import them into Jira/GitHub Issues). Each is sized (S/M/L), with AC and dependencies.

**Epic E2: Auth Lifecycle (sample)**

* **US-E2-01 (M):** *As a user* I want sign-out to fully clear my session so returning to /engagements requires a fresh Microsoft login.

**AC:** (GWT) Given an authenticated session, when I click sign-out, then navigating to /engagements must redirect to Microsoft (no auto-login).

**Deps:** NextAuth routes, cookie config, AAD logout URIs.

* **US-E2-02 (S):** *As a user* I must never see 0.0.0.0:3000 / localhost in any prod sign-in/out redirects.

**AC:** All redirects show Container Apps host only; PR includes tests.

**Epic E3: Contracts (sample)**

* **US-E3-01 (M):** *As a consultant* I need /api/proxy/presets to list supported models (e.g., cscm-v3, cyber-for-ai) so I can start quickly.

**AC:** List returns 2xx with expected ids; UI table non-empty.

* **US-E3-02 (M):** *As a consultant* I need create→read assessment to work so the detail page loads without error.

**AC:** 201 on create, 200 on subsequent get; UI shows detail; no “Failed to get assessment.”

**Epic E4: Orchestrator (sample)**

* **US-E4-01 (M):** *As a user* I want chat to route “analyze this doc” to Doc Analyzer and save results in the conversation context.

**AC:** Chat turn shows “analyze doc” result; run log recorded; context persisted.

(…continue listing stories for each Epic as needed…)

**4) UAT Scripts (per sprint)**

* **Auth UAT:** Visit /signin → click AAD → Microsoft prompt → back to /engagements; sign-out → /signin; repeat AAD → ensure **no auto-login**.
* **Contracts UAT:** Presets table non-empty; upload returns 201 → visible; create assessment → open detail; engagements list 2xx; admin/status 200.
* **Chat UAT:** “Analyze doc” works; “Transcribe audio” works; conversation store has run logs.
* **Scoring UAT:** Score computed; reasons and missing-evidence prompts displayed.
* **Roadmap UAT:** Deck generated; Jira issues created idempotently.
* **RAG UAT:** Answers show citations; contradictions listed.
* **Consent UAT:** Audio ingest requires consent; audit entry visible.

**5) Governance & Metrics**

* **Delivery metrics:** Sprint burn-up, lead time for changes, change failure rate, mean time to restore.
* **Quality metrics:** Test coverage trends, UAT pass rate, Actions failure categories.
* **Security metrics:** High/Critical findings (#/age), key rotations, dependency freshness.
* **SLOs:** API error rate, P95 latency, Uptime; alerting thresholds.

**6) Final Notes**

* Keep changes **small** and **auditable**; show value **every sprint**.
* Each sprint ends with **deployment + SHA proof + UAT Gate + demo**.
* No workarounds. When a trade-off is needed, update the **architecture doc** first and provide migration notes.
* Keep the **Development Guidelines** and **Team Playbook** as the first point of reference before every change.