ODA3 INSTITUTE

Practitioner Cheat Sheet

AI Security Assurance Framework

DocID: ODA3-2026-07-CHT-SEC-006
Classification: Public

CHT-SEC-006: AI Security Assurance Framework

ODA3 Institute | *Practitioner Cheat Sheet*

Field	Value
Classification	Public — Practitioner Reference
Intended Audience	CISOs, AI Security Architects, Security Assurance Teams, Internal Audit, AI Governance Leads, Enterprise Risk Functions
Evidence Confidence	Moderate (T2–T3 weighted; see tier tags)
Review Cycle	6 months
Framework Cross-References	Maps to UAIF™ v1.0 (classification), GAISSF™ v1.0 (assurance), AI-IRF™ v1.0 (incident response)
Related Publications	CHT-SEC-001 through CHT-SEC-005

Why This Matters

Many organizations have implemented AI security controls but cannot confidently answer a simple question: how do we know these controls actually work?

Traditional security assessments often stop at confirming a control exists. AI adds several complications that make that insufficient: model behavior is probabilistic, datasets and prompts change continuously, adversaries adapt, agents act autonomously, and much of the stack depends on external foundation models nobody in the building can directly inspect. A point-in-time answer to "does this control exist" says almost nothing about whether it still works next month.

AI Security Assurance is not another audit. It is the continuous discipline of building justified confidence — backed by evidence, not assertion — that AI security objectives are actually being met throughout the system's life, not just on the day someone checked.

1. What Assurance Actually Answers

Assurance is often conflated with governance, compliance, and audit. Each is necessary, but each answers a different question — and mistaking one for another is the single most common failure mode in this space.

Discipline	Primary Question	Typical Output
Governance	What should be done?	Policies, standards, decision rights
Risk Management	What risks exist?	Risk register, treatment plans
Security Engineering	How is the system protected?	Security controls, architecture
Compliance	Are obligations satisfied?	Compliance reports, attestations
Audit	Were controls followed at a point in time?	Audit findings
Assurance	Can we justify confidence that controls remain effective, continuously?	Evidence-based confidence assessment

> **The practitioner error:** treating a passed audit or a signed compliance attestation as proof of assurance. Both are legitimate inputs to assurance — neither is assurance on its own [T3 — consistent with general control-testing literature; not a claim specific to AI].

2. Principles of Assurance

Before describing domains, evidence, or maturity levels, it's worth stating the properties that make an assurance program credible in the first place. An assurance program missing any one of these produces conclusions that look rigorous but don't hold up under scrutiny.

- **Independence.** The function reaching an assurance conclusion should not be the same function that built or operates the control being assessed. Self-assessment is a legitimate evidence input; it is not, by itself, assurance (see Section 13).
- **Objectivity.** Conclusions should follow from evidence, not from the reputation of the team or the plausibility of the design. A well-regarded engineering team's untested claim carries the same evidentiary weight as anyone else's untested claim: none.
- **Repeatability.** An assurance activity should produce comparable results if repeated by a different qualified assessor under similar conditions. A test only one specific person can perform, with undocumented judgment calls throughout, is not repeatable and weakens confidence in its result.
- **Traceability.** Every assurance conclusion should be traceable back to specific evidence artifacts — which test, run when, against which system version, reviewed by whom. A conclusion that can't be traced back to its source evidence can't be defended when challenged.
- **Continuous improvement.** Assurance findings should feed back into governance, architecture, and controls — not sit in a report nobody revisits. A program identifying the same gap across three consecutive review cycles without remediation is not providing assurance; it's documenting an unaddressed problem.

These five principles apply across every domain, activity, and evidence type described in the rest of this cheat sheet — they're the test for whether an assurance activity is real or performative.

3. Assurance vs Related Disciplines (Detail)

Building on Section 1, it's worth being explicit about where each discipline's output ends and assurance begins.

Question	Discipline
Does a security policy exist?	Governance
Was the control implemented?	Security Engineering
Does the control satisfy regulatory obligations?	Compliance
Does the control consistently work in practice?	Assurance
Can we demonstrate effectiveness using evidence?	Assurance

4. The Assurance Lifecycle

Unlike a traditional audit cycle, assurance is a loop, not a sequence with an end date.

```
Security Objectives
```

```
        |
Risk Assessment
        |
Architecture Review
        |
Control Implementation
        |
Evidence Collection
        |
Validation & Testing
        |
Continuous Monitoring
        |
Independent Assurance
        |
Management Review
        |
Continuous Improvement ---> (back to Security Objectives)
```

Every stage after "Control Implementation" should be recurring, not one-time. A system that has been through this loop once has been assessed once — it has not yet accumulated assurance, which requires the loop to keep running.

5. Five Assurance Domains

A mature assurance program evaluates five complementary domains. Focusing on only one — usually Control Assurance — is the most common way organizations end up with a false sense of security.

Domain	Primary Objective
Governance Assurance	Confirm governance structures actually support secure AI operations
Architecture Assurance	Validate that the system's design enforces the intended trust boundaries
Control Assurance	Verify security controls operate as intended, continuously
Operational Assurance	Confirm secure day-to-day operations across the AI lifecycle
Evidence Assurance	Demonstrate effectiveness through evidence that is objective and traceable

5.1 Governance Assurance

Asks whether organizational governance is actually capable of enabling secure AI operations — not whether a policy document exists, but whether it functions.

Representative questions: Are security responsibilities explicitly assigned? Is AI security integrated into enterprise governance rather than run in a silo? Are risk-acceptance decisions documented with a named owner? Are AI systems formally inventoried?

Representative evidence: governance policies, RACI matrices, risk registers, committee minutes, approval records.

5.2 Architecture Assurance

Asks whether the system's architecture (see CHT-SEC-005) enforces the security objectives it was designed for — trust boundaries, identity architecture, model isolation, retrieval architecture, tool security, network segmentation, and third-party dependencies. This evaluates design quality before an incident forces the question.

5.3 Control Assurance

Confirms that implemented controls — authentication, authorization, prompt security, context validation, retrieval authorization, encryption, logging, monitoring, rate limiting, output filtering — continue to function effectively, not merely that they were configured correctly once.

5.4 Operational Assurance

Assesses whether secure operations are sustained across the lifecycle: incident response, change management, model deployment, configuration management, key management, vulnerability management, third-party oversight.

5.5 Evidence Assurance

The foundation the other four domains depend on. Without objective evidence, none of the other domains' conclusions can be justified — they become assertions. Evidence Assurance is developed fully in Sections 7–9 below.

6. Assurance Maturity Model

Level	Description
Level 1 — Ad Hoc	Informal security activities with limited evidence
Level 2 — Repeatable	Security processes documented and periodically performed
Level 3 — Defined	Standardized assurance methodology applied consistently across AI systems
Level 4 — Managed	Metrics-driven assurance supported by continuous monitoring
Level 5 — Optimized	Continuous assurance integrated into governance, engineering, and enterprise risk management

Progression through these levels comes from improving governance, evidence quality, and automation — not simply from buying more security tooling.

7. The Assurance Evidence Model

An organization should be able to answer not just what controls exist but how it knows those controls remain effective. Every significant assurance conclusion should be traceable to evidence — this is the Traceability principle from Section 2 in practice.

Characteristics of high-quality evidence: objective, repeatable, independently verifiable, attributable, traceable, timely, and protected against unauthorized modification. Weak evidence undermines confidence regardless of how sophisticated the underlying control is.

Evidence Source	Examples
Governance	Policies, standards, committee approvals, risk-acceptance records
Architecture	Architecture diagrams, trust-boundary documentation, threat models
Technical Controls	Configuration baselines, IAM policies, firewall rules, model access policies
Monitoring	SIEM alerts, prompt monitoring, model telemetry, retrieval logs
Testing	Red-team reports, penetration testing, adversarial evaluations
Operations	Incident records, change requests, deployment approvals

Evidence Source	Examples
Compliance	Audit reports, regulatory assessments, control attestations
External	Independent assessments, supplier assurance reports, certifications where applicable

8. Security Evidence Hierarchy

Not all evidence carries equal weight. Confidence generally rises as evidence becomes more operational, observable, and independently validated.

```
Independent Assessment          <- strongest
        ^
Red-Team Results
        ^
Continuous Monitoring
        ^
Technical Testing
        ^
Configuration Evidence
        ^
Documentation                   <- weakest
```

Evidence Type	Confidence
Policy document	Low
Architecture review	Low–Moderate
Configuration review	Moderate
Security testing	Moderate–High
Continuous monitoring	High
Independent assessment	Very High

> **Practical principle:** several independent, moderate-quality sources generally provide stronger assurance than reliance on a single high-quality artifact. A policy document plus a configuration review plus continuous monitoring evidence beats one excellent red-team report standing alone.

9. Assurance Activities Matrix

Different assurance activities answer different questions; a mature program combines several.

Assurance Activity	Primary Objective	Typical Frequency
Architecture Review	Validate secure design	Major changes
Threat Modeling	Identify security risks	Design and significant updates
Control Review	Verify implementation	Periodic
Configuration Review	Validate secure configuration	Periodic
Vulnerability Assessment	Identify weaknesses	Regular
AI Red Teaming	Evaluate adversarial resilience	Periodic

Assurance Activity	Primary Objective	Typical Frequency
Penetration Testing	Assess exploitable weaknesses	Periodic
Monitoring Review	Confirm operational visibility	Continuous
Incident Review	Improve controls following events	After significant incidents
Independent Assessment	Provide objective assurance	Scheduled

10. Verification vs Validation vs Testing

Three more terms that get flattened into "we checked it" and shouldn't be.

Verification	Validation
Did we build the control correctly?	Does the control achieve its intended objective?
Implementation-focused	Outcome-focused
Configuration review	Operational effectiveness
Static assessment	Dynamic assessment
Engineering activity	Assurance activity

> **Worked example:** prompt filtering is configured exactly to specification. Verification confirms the filter was implemented as designed. Validation confirms the filter actually blocks malicious prompts under realistic operating conditions — a materially different and stronger claim.

Testing	Assurance
An individual activity	A continuous discipline
Measures technical behavior	Evaluates confidence across governance, controls, operations, and evidence
Often point-in-time	Lifecycle-oriented
Produces findings	Produces justified confidence

Testing — penetration testing, adversarial testing, red teaming, vulnerability assessment, configuration validation — contributes evidence. It is not, by itself, assurance. Assurance is what happens when those results get integrated with governance, operational performance, and continuous monitoring over time.

11. Continuous Assurance

AI systems change constantly — model updates, prompt changes, retraining, new data, infrastructure changes, regulatory developments, emerging attack techniques. Assurance has to run on the same continuous cadence or it falls permanently behind.

```
Observe -> Measure -> Evaluate -> Improve -> (back to Observe)
```

Representative continuous assurance activities: continuous control monitoring, model drift monitoring, prompt injection detection, policy compliance monitoring, retrieval authorization validation, security telemetry review, configuration drift detection, automated evidence collection.

12. AI Security Metrics by Domain

Metrics should measure whether capabilities are effective, not just whether activities occurred.

- **Governance:** AI systems with assigned owners; policies reviewed on schedule; outstanding risk-acceptance decisions; governance exceptions.
- **Architecture:** systems with documented trust boundaries; AI services behind an AI Security Gateway (CHT-SEC-005, Pattern 1); architectures reviewed before deployment; high-risk architecture findings.
- **Control:** authentication success rates; authorization failures; prompt injection detection rate; retrieval authorization violations; policy enforcement failures.
- **Operational:** mean time to detect (MTTD); mean time to respond (MTTR); AI-involved security incidents; configuration drift events; third-party security events.
- **Evidence:** controls with current evidence; automated evidence coverage; independent assessments completed; evidence review exceptions; expired evidence artifacts.

> **Practitioner note:** "number of policies written" and "number of scans completed" are activity counts, not assurance metrics — they say nothing about whether risk actually went down.

13. Assurance Responsibilities

Assurance is a shared responsibility across functions — no single function should reach an assurance conclusion alone, per the Independence principle in Section 2.

Function	Primary Assurance Responsibility
Board	Oversight, assurance expectations, risk appetite
Executive Leadership	Resource allocation, accountability
AI Governance Lead	Governance assurance coordination
CISO	Technical security assurance
Engineering	Control implementation and operational evidence
Enterprise Risk Office	Enterprise integration and reporting
Compliance	Regulatory assurance
Legal	Legal interpretation and review
Internal Audit	Independent assurance
Business Owner	Acceptance of residual business risk

A recurring failure pattern: the same team that builds a control also being the sole voice declaring it assured. Independent review (Internal Audit, or an external party) should validate any assurance conclusion that carries material weight.

14. Assurance Confidence Model

Rather than a binary pass/fail, assurance conclusions should be expressed as a confidence level.

Confidence Level	Characteristics
Very Low	Limited documentation, minimal evidence
Low	Controls implemented but effectiveness uncertain
Moderate	Multiple evidence sources with periodic validation
High	Continuous monitoring supported by independent review
Very High	Multiple independent evidence sources with mature governance and continuous assurance

14.1 Assurance Confidence Is Not Risk Level

These two get conflated constantly, and the conflation causes bad decisions in both directions — so it's worth stating the distinction explicitly rather than leaving it implied.

Risk level describes the potential impact and likelihood of something going wrong — how bad it would be, and how likely, independent of how well anyone has checked. Assurance confidence describes how much justified confidence exists that the controls managing that risk actually work, based on evidence quality and independence.

A system can be high risk with high assurance confidence: a critical, high-impact AI system with continuous monitoring, independent red-teaming, and strong evidence across all five domains. The risk is inherently large, but the organization has real, well-evidenced confidence it's being managed.

A system can also be low risk with low assurance confidence: an internal, low-stakes tool nobody has ever tested or reviewed. The potential impact is small, but the organization genuinely doesn't know whether its (probably minimal) controls work — the confidence level is low regardless of the low stakes.

	Low Assurance Confidence	High Assurance Confidence
High Risk	Dangerous: high-stakes system, unproven controls — the worst quadrant, and the one most likely to be mistaken for "fine" if no one is checking	Defensible: high-stakes system with continuous, independent evidence that controls work
Low Risk	Common and often acceptable: low-stakes system, unproven controls — usually a reasonable place to under-invest in assurance	Arguably over-assured: resources spent proving low-stakes controls work might be better spent elsewhere

> **Practical implication:** assurance investment should track risk level (CHT-SEC-004, Section 5 covers risk prioritization), but the two numbers are never a substitute for each other. Reporting "this system is high assurance" says nothing about whether it's also high risk, and reporting "this is low risk" says nothing about whether anyone has actually verified that.

15. Evidence Quality Matrix

A practical rubric for evaluating any single piece of evidence before it's used to support a conclusion.

Characteristic	Low Quality	Moderate Quality	High Quality
Objectivity	Opinion-based	Partially measurable	Fact-based and independently verifiable

Characteristic	Low Quality	Moderate Quality	High Quality
Timeliness	Historical	Periodically updated	Current or continuously collected
Repeatability	Ad hoc	Documented process	Consistently reproducible
Traceability	Limited	Partial chain of custody	Complete provenance maintained
Independence	Self-assessed	Peer reviewed	Independently validated
Automation	Manual	Semi-automated	Automated evidence collection
Integrity	Editable without controls	Protected by process	Cryptographically or procedurally protected

16. Assurance Decision Matrix

Situation	Assurance Decision
Policy exists but no operational evidence	Insufficient assurance
Controls implemented but never tested	Limited assurance
Controls tested once during deployment	Moderate assurance
Controls continuously monitored	High assurance
Continuous monitoring plus independent assessment	Very High assurance
Evidence incomplete or outdated	Reassess before relying on conclusions
Significant architecture change without reassessment	Previous assurance no longer reliable

```
Security Objective
        |
Evidence Available?
   No ----> Insufficient Assurance
   Yes ---> Is it independently validated?
              No  ----> Moderate Assurance
              Yes ---> High Assurance
```

> **Governing principle:** an assurance conclusion should always reflect the strength of the evidence actually available — never an assumption about how effective a control is likely to be.

17. Mapping to ODA3's Frameworks

ODA3 Framework	Role in Assurance
UAIF™ v1.0	Classifies the incident when an assurance gap is exploited — e.g., a control assessed as "High Assurance" that is later bypassed classifies differently than one honestly logged as "Insufficient Assurance" and accepted as residual risk.
GAISSF™ v1.0	Is the assurance layer this cheat sheet describes in practice — GAISSF's Foundational/Operational/Optimized Certification tiers correspond directly to increasing rigor across the Confidence Model (Section 14) and Evidence Quality Matrix (Section 15). Foundational Attestation expects Moderate assurance evidence across the 52 canonical controls; Optimized Certification expects Very High assurance with independent, continuous evidence.

ODA3 Framework	Role in Assurance
AI-IRF™ v1.0	Governs the response when an assurance conclusion turns out to have been wrong — the Assurance Decision Matrix's "reassess" triggers (Section 16) are designed to route into AI-IRF-aligned identification and lessons-learned phases, not just a quiet correction.

Safe-harbor note: this section describes how assurance concepts relate to ODA3's frameworks in principle. It does not assert that any specific organization holds a GAISSF™ tier or is UAIF-classified — tier and classification determinations require a formal ODA3 assessment.

17a. External Framework Landscape (Reference, Not Endorsement)

The assurance concepts in this cheat sheet aren't ODA3 inventions — they echo structure already emerging across regulatory and industry guidance. These are cited for orientation only; ODA3 Institute is not affiliated with NIST, ISO, IEC, the European Union, MITRE, CSA, OWASP, or ETSI, and mapping a concept to a framework below does not assert certification, compliance, or equivalence.

Framework	Type	Relevance to Assurance
NIST AI RMF 1.0	Voluntary risk-management framework (Govern, Map, Measure, Manage)	The Measure function maps closely to this cheat sheet's Continuous Assurance (Section 11) and Evidence Model (Section 7) [T2 — publicly documented framework]
ISO/IEC 42001:2023	Certifiable AI management system standard	The only framework here that issues a certification comparable in structure to GAISSF™; its Clause 9 performance-evaluation requirements parallel this cheat sheet's evidence and continuous-monitoring expectations [T2]
EU AI Act	Binding EU regulation for high-risk systems	Requires documented risk management and post-market monitoring — directly relevant to the Continuous Assurance and Reassurance concepts in Sections 11 and 16 [T2 — regulatory text is public record; verify current enforcement timelines against EU sources]
MITRE ATLAS	Adversarial technique knowledge base for AI/ML systems	Provides the threat vocabulary for Control Assurance (Section 5.3) adversarial testing activities [T2]
CSA AI Controls Matrix (AICM)	Vendor-agnostic controls framework	A candidate source of control baselines against which Control Assurance evidence (Section 5.3, Section 7) could be organized [T3 — community-maintained, versioned frequently]
OWASP AISVS / AI Testing Guide	Testable security requirements and testing methodology	A candidate source for the specific test cases behind this cheat sheet's Control Assurance and Assurance Activities Matrix (Section 9) [T3]
ETSI EN 304 223	Cybersecurity standard across 5 lifecycle phases	Its lifecycle framing (design through end-of-life) parallels the lifecycle stages referenced throughout this cheat sheet [T3 — recently published; adoption timeline not yet independently verified]

Practitioner takeaway: none of these frameworks use the term "assurance" identically to how it's

defined in Section 1 — some (NIST, EU AI Act) treat it as embedded within risk management, others (ISO 42001) as a certification outcome, and others (MITRE, CSA, OWASP) as a technical testing concern. This cheat sheet's five-domain model (Section 5) is one way to reconcile those perspectives into a single practitioner vocabulary, not a claim that the frameworks themselves use this exact structure.

18. Assurance for Agentic AI Specifically

Agentic systems (see CHT-SEC-005, Patterns 8, 11, 12) raise the assurance bar because the thing being assured now includes autonomous decision-making and action-taking, not just text generation. All five domains from Section 5 still apply — agentic systems simply demand more from Architecture and Control Assurance specifically.

Additional Assurance Requirement	Why Agentic Systems Need It
Tool-scope verification	Confirming an agent's actual runtime permissions match its documented least-privilege scope, not just its configuration file
Action reversibility testing	Verifying that high-impact actions actually route through Human Approval (CHT-SEC-005, Pattern 6) rather than assuming the workflow diagram reflects the deployed system
Session isolation testing	Confirming one agent session cannot access another's memory or credentials, not just that the architecture diagram shows separation
Chained-action assurance	Testing multi-step agent workflows end-to-end, since assurance of each individual tool call doesn't guarantee assurance of the full chain

19. Third-Party and Vendor Assurance

Most organizations run at least one AI system built substantially on third-party models, APIs, or data — assurance obligations don't stop at the organizational boundary, and Section 2's Independence principle applies just as much to a vendor's self-reported assurance as to an internal team's.

- **Vendor control attestation.** Request evidence — not just marketing claims — that the vendor's own security controls are tested, not merely documented.
- **Shared responsibility mapping.** Document explicitly which assurance activities the vendor performs versus which remain the deploying organization's responsibility; assuming the vendor "handles security" is a common gap.
- **Independent verification where feasible.** For high-exposure systems, don't rely solely on vendor-provided assurance evidence; commission independent testing of the integration points you control.
- **Reassessment on vendor model updates.** A vendor's silent model version change is functionally equivalent to an internal retraining event (Section 16) and should trigger the same reassurance cycle.

20. Real-World Assurance Pattern Reference

Included to illustrate assurance gaps as a pattern, not to assess any specific organization.

Pattern (general, not case-specific)	Assurance Gap	What Would Have Closed It
A model passes pre-deployment red-	No reassurance trigger tied to	Section 16's reassessment

Pattern (general, not case-specific)	Assurance Gap	What Would Have Closed It
teaming, then is fine-tuned post-launch with no re-test	model change	trigger for significant change
An agent's documented tool scope is narrower than its actual runtime permissions	Configuration drift between design and deployment, uncaught by documentation review alone	Section 18's tool-scope verification
A vendor silently updates an underlying foundation model	No shared-responsibility mapping for reassurance ownership	Section 19's reassessment-on-vendor-update practice

21. Assessment Readiness Checklist

- ☐ AI systems formally inventoried
- ☐ Security objectives documented
- ☐ AI risk assessments completed
- ☐ Architecture diagrams maintained and current
- ☐ Threat models documented
- ☐ AI Security Gateway implemented where applicable (CHT-SEC-005)
- ☐ Security controls documented with named owners
- ☐ Security monitoring operational, not just configured
- ☐ AI-specific telemetry collected (prompts, retrieval, tool calls — not just infrastructure)
- ☐ AI incident response procedures established
- ☐ Red-team exercises completed against the deployed system
- ☐ Independent reviews scheduled, not ad hoc
- ☐ Residual risks documented with named owners and review dates
- ☐ Risk acceptance formally approved, not informally tolerated
- ☐ Evidence repository maintained centrally, not scattered across tickets and chat threads
- ☐ Governance reviews conducted on a defined periodic cadence
- ☐ Third-party AI services assessed against the same expectations as internal systems
- ☐ Assurance confidence levels reported to leadership, distinguished explicitly from risk level (Section 14.1)

22. Common Organizational Mistakes

- **Confusing compliance with assurance.** Passing a compliance assessment shows alignment with a specific requirement — it doesn't show the control works under real conditions. Use compliance as one evidence input, not the whole answer.
- **Treating assurance as an annual audit.** AI systems change continuously; an annual snapshot cannot sustain confidence for the other eleven months.
- **Collecting evidence without objectives.** Extensive logs and dashboards without a defined question they're meant to answer produce noise, not assurance.
- **Measuring activity instead of effectiveness.** Policy counts and completed-scan counts don't show risk actually went down.
- **Relying on a single evidence source.** One excellent penetration test or one dashboard is only partial confidence — combine governance, technical, operational, and independent evidence.

- **Ignoring architectural change.** A new model, retrieval pipeline, agent, or third-party integration changes the assurance picture; reassess after significant changes, not on the next calendar date.
- **Treating assurance as a security-team-only responsibility.** Governance, engineering, risk, legal, compliance, business owners, and internal audit all contribute — see Section 13.
- **Conflating assurance confidence with risk level.** Reporting one without the other, or treating "high confidence" as if it means "low risk," misleads decision-makers — see Section 14.1.

Notably Absent

This cheat sheet does not include:

- Proprietary assurance evidence, client assessment data, or vendor-specific results (ODA3 Institute, founded March 2026, holds no such datasets)
- A step-by-step GAISSF™ assessment methodology — Section 17 describes how assurance concepts relate to GAISSF™ tiers, not the assessment procedure itself
- Assertions that any organization holds a specific GAISSF™ tier or UAIF classification — these require a formal ODA3 assessment
- Specific tooling or vendor recommendations for evidence collection, monitoring, or red-teaming
- A quantitative scoring formula for the Confidence Model (Section 14) — confidence levels here are qualitative and directional, not a validated numeric instrument
- A complete audit or compliance framework — this cheat sheet addresses assurance specifically and does not restate CHT-SEC-004's risk taxonomy or CHT-SEC-005's architecture patterns beyond what's needed for cross-reference

Five Priority Actions

- **1. Define assurance objectives before collecting evidence.** Decide which confidence questions the organization actually needs answered, then work backward to what evidence would answer them.
- **2. Establish an evidence governance process.** Standardize how evidence is collected, protected, reviewed, retained, and traced — not scattered across whichever tool happened to log it.
- **3. Move from point-in-time reviews to continuous assurance.** Pair periodic assessments with continuous monitoring and automated evidence collection (Section 11).
- **4. Integrate assurance across governance, engineering, and operations.** A unified program beats five disconnected review processes that never compare notes.
- **5. Report assurance as confidence levels, explicitly separate from risk level.** Give leadership the Section 14 and 14.1 vocabulary — strengths, limitations, residual risk, and what would raise the confidence level — instead of a binary green light that quietly implies risk is also low.

This cheat sheet maps to GAISSF™ v1.0 assurance guidance, UAIF™ v1.0 classification categories, and AI-IRF™ v1.0 incident response principles. It does not constitute certification, compliance confirmation, or endorsement under any referenced framework.

ODA3 Institute — Where AI Governance meets Operational Reality.