CXO INTELLIGENCE
Leadership in the Age of AI

CXO INTELLIGENCE SERIES ON GOVERNANCE

SINGAPORE MODEL AI GOVERNANCE FRAMEWORK FOR AGENTIC AI Consolidated Checklist

Every control with an implementation strategy, priority and effort rating

Model AI Governance Framework for Agentic AI (v1.5), mapped to ISO/IEC 42001

Prashant Akhawat | akhawat.com
CXO Intelligence Series | Subscribe on LinkedIn

How to Use This Checklist

This is a consolidated, working checklist of every control in Singapore's Model AI Governance Framework for Agentic AI (v1.5). Each row pairs the control with a concrete implementation strategy, a priority and an effort rating, and the closest ISO/IEC 42001 reference so you can align to the management system standard as you go.

Reading the columns

- **Control:** the control ID and short name.
- **Description:** what the control requires, as an outcome.
- **Implementation Strategy:** concrete, in-depth steps to put the control in place.
- **Priority:** Critical, High or Medium, to help you sequence the work.
- **Effort:** High, Medium or Low, an indicative implementation effort.
- **ISO 42001 ref:** the closest clause or Annex A control for management-system alignment.

Suggested sequencing

Start with the Critical, Low-effort controls for quick wins, then the Critical, higher-effort controls, then work down through High and Medium. Priority reflects risk reduction; effort reflects implementation cost, so the two together guide your roadmap.

Scope note: *The MGF content is taken from the published agentic-AI framework in your project. ISO/IEC 42001 references are drawn from the standard's structure for alignment and are paraphrased, not quoted, because the standard is copyrighted. Priority and effort are indicative and should be calibrated to your context. Educational tool, not legal advice.*

The Consolidated Checklist

Priority: Critical High Medium **Effort:** High Medium Low

Control	Description	Implementation Strategy	Priority	Effort	ISO 42001 ref
Foundation - Agentic Risk Identification (MGF Section 1.2)					
SG-0.01 **Identify risk by source (agent components)**	Recognise that each agent component is a new source of risk: planning/reasoning (hallucination, semantic misalignment, plan drift), tools (wrong/non-existent tool calls, injection-driven exfiltration), and protocols (compromised or untrusted MCP servers).	Stand up a component-level risk register for every agent. Enumerate risks across all eight core components the framework identifies: model, instructions, memory, planning and reasoning, tools, protocols, controls, and logging and monitoring. Record inherited software and LLM risks, and review at design and on any architecture change.	Critical	Medium	Cl.6.1.2; A.5.2
SG-0.02 **Assess types of real-world harm**	Assess the categories of harmful outcome an acting agent can cause: erroneous actions, unauthorised actions, biased/unfair actions, data breaches, and disruption to connected systems.	Create a harm-type matrix mapping each use case to the five harm categories (erroneous, unauthorised, biased, data breach, system disruption). For each, capture a concrete scenario and severity to drive control selection and approval gates.	Critical	Low	A.5.4; A.5.2
SG-0.03 **Assess speed & volume risk to oversight**	Recognise that agent decision speed can outpace real-time oversight, and that continuous human approval itself induces automation bias and alert fatigue.	Assess whether human oversight can keep pace with the agent's decision speed. Where it cannot, shift from continuous manual approval toward deterministic controls and automated monitoring, and set alert thresholds.	High	Low	A.9.2; A.6.2.6
SG-0.04 **Assess cascading & compounding effects**	Assess how an early error (e.g. a hallucinated figure) can propagate and amplify across later steps, producing outsized downstream impact.	Trace multi-step workflows for amplification points where an early error compounds. Insert validation checkpoints or human review between steps that could cascade, especially for high-impact chains.	High	Medium	A.5.5; Cl.6.1.2
SG-0.05 **Assess multi-agent & systemic risks**	Assess risks unique to multi-agent systems: agent sprawl, collaborative failures (miscoordination, conflict, collusion), and unpredictable emergent behaviour from interacting non-deterministic agents.	For multi-agent systems, first identify which of the three interaction patterns applies (sequential, supervisor, or swarm), then model the interaction failure modes (miscoordination, conflict, collusion, emergent behaviour). Use taint tracing across agents, limit shared memory, monitor agent-to-agent exchanges, and manage agents centrally to curb sprawl.	High	High	A.5.5; A.4.2
SG-0.06 **Assess cross-system / cross-boundary risk**	Recognise that when agents cross system or organisational boundaries (especially without white-box access to external systems), outcomes are harder to test and anticipate.	Identify every cross-system and cross-organisation interaction. Require transparency from external parties, and contain or sandbox cross-boundary actions that cannot be exhaustively tested.	Medium	Medium	A.10.3; A.6.2.4
SG-0.07 **Define action-space & autonomy level per agent**	Characterise each agent on the action-space (small to large) and autonomy (human-approves-each-step, operates-under-SOP, operates-with-post-hoc-audit) spectrum, as this determines inherent risk.	Characterise each agent explicitly on the action-space and autonomy spectrum, and record the chosen level. Tie that level to the governance pathway and the required control intensity so higher autonomy triggers heavier controls.	Critical	Low	A.6.2.2; Cl.6.1.2
Dimension 1 - Assess & Bound the Risks Upfront					
SG-1.01 **Agentic use-case suitability assessment**	Evaluate whether a use case suits an agent at all, weighing risk (likelihood x impact) vs benefit; some tasks better as deterministic workflows.	Introduce a documented go or no-go gate before any build. Score likelihood and impact, compare against a deterministic-workflow alternative, and record the decision and rationale in the use-case register. No build proceeds without a passed gate.	Critical	Low	Cl.6.1.2; A.5.2
SG-1.02 **Impact factor analysis**	Assess severity factors: domain error tolerance, criticality of supported processes, sensitive-data access, external-system access.	Add an impact-factor rating step to the intake process: domain criticality, sensitive-data access, external-system access and reversibility. Flag high-	Critical	Low	A.5.2; A.5.4

Control	Description	Implementation Strategy	Priority	Effort	ISO 42001 ref
		impact combinations for stricter controls and escalation to the governance committee.			
SG-1.03 **Likelihood factor analysis**	Assess probability factors: autonomy level, task complexity, external/untrusted exposure, third-party provision, system complexity.	Add a likelihood-factor rating covering autonomy, task complexity, external exposure, third-party provision and system complexity. Combine with impact to set a risk tier that drives monitoring and human-in-the-loop intensity.	Critical	Low	Cl.6.1.2
SG-1.04 **Threat modelling & taint tracing**	Systematically identify attack paths (memory poisoning, tool misuse, privilege compromise, indirect prompt injection) and trace untrusted-data flows.	Adopt an AI-adapted threat-modelling method and run it at design for every agent. Map untrusted-data flows with taint tracing, and keep the model live, updated on architecture, tool or model change, cross-referenced to OWASP agentic threats.	High	High	A.6.2.4; A.8.3
SG-1.05 **Least-privilege limits on tools & system access**	Give each agent only the minimum tools and data access; structure agents around functional boundaries.	Define and enforce a least-privilege access policy per agent at the tool layer. Remove unneeded tools, split broad agents by function, and review permissions periodically. Default to the minimum scope needed for the task.	Critical	Medium	A.4.4; A.6.2.3
SG-1.06 **Bound agent autonomy with SOPs**	Constrain agents to defined SOPs for process-driven tasks rather than free-form step definition.	For process-driven tasks, encode the required procedure as enforced workflow steps rather than prompt suggestions. Reserve open-ended autonomy for low-risk, low-impact tasks, and document the SOP each agent must follow.	High	Medium	A.6.2.2
SG-1.07 **Limit agent area of impact / containment**	Design mechanisms to take agents offline and confine scope on malfunction; sandbox high-risk tasks like code execution.	Run high-risk agents in sandboxes with restricted network and data access, and provide a tested offline or kill mechanism. Contain blast radius by default and expand scope only after validation.	High	High	A.6.2.6; A.4.5
SG-1.08 **Prefer deterministic, bound-by-design limits**	Impose deterministic controls (e.g. access controls preventing a tool call) rather than prompt instructions; layer monitoring where limits are weaker.	Implement limits at the system and tool layer, not the prompt layer. For any control that must remain non-deterministic, add compensating monitoring or human-in-the-loop so a bypass is detected.	High	Medium	A.6.2.3
SG-1.09 **Unique, verifiable agent identity**	Each agent has its own unique, cryptographically verifiable identity.	Issue a unique, cryptographically verifiable identity per agent instance. Avoid shared or human-borrowed credentials, and use standards such as OAuth 2.1 with MCP where available.	High	High	A.6.2.6
SG-1.10 **Accountable & capacity-differentiated identity**	Tie each identity to an accountable human/dept; record the capacity in which the agent acts (independent vs on-behalf-of).	Map every agent identity to an accountable human or department, and log the capacity in which the agent acts, whether autonomously or on a user's behalf. Record delegation chains for sub-agents.	High	Medium	A.3.2; A.6.2.6
SG-1.11 **Central identity catalogue (prevent sprawl)**	Issue and track all agent identities/permissions centrally to inventory agents, detect anomalies, and revoke unused identities.	Stand up a central agent registry that issues and tracks all agent identities and permissions. Require registration before deployment, review periodically, and de-provision dormant or orphaned agents to prevent sprawl.	High	Medium	A.4.2
SG-1.12 **Third-party agent platform containment**	Default-restrict vendor/SaaS agents to their own ecosystems rather than free action across enterprise data/systems.	Set a default policy that vendor and SaaS agents operate only within their own ecosystem. Require review and testing before enabling any cross-platform action, and register third-party agents centrally.	Medium	Medium	A.10.3
SG-1.13 **Calibrate governance to level of agency**	Create risk-calibrated governance pathways by level of agency, with heavier review and runtime enforcement for higher autonomy.	Define tiers of agency, each with a matching governance pathway: lightweight for low tiers, impact assessment for the middle, and architecture review plus runtime policy enforcement for the highest autonomy.	High	Medium	Cl.6.1.3; A.5.5

Control	Description	Implementation Strategy	Priority	Effort	ISO 42001 ref
SG-1.14 **Residual-risk evaluation & acceptance**	Formally evaluate whether remaining risk is tolerable and can be accepted before deployment.	Document residual risk per use case against risk appetite, and require named-owner sign-off to accept it before deployment. Re-evaluate on any material change or incident.	Critical	Low	Cl.6.1.3; Cl.8.3
Dimension 2 - Make Humans Meaningfully Accountable					
SG-2.01 **Map the agentic AI value chain**	Identify all actors: model developers, platform providers, system providers, deployers, tooling providers, end users.	Document the agentic value chain for each agent, noting which roles your organisation plays. Use the map to allocate obligations across model developers, platform and system providers, deployers, tooling providers and users.	High	Low	A.3.2; A.10.2
SG-2.02 **Clear allocation of responsibilities (internal)**	Establish internal accountability chains across the agent lifecycle.	Publish a lifecycle RACI naming the use-case owner, risk team, builder and reviewers, and their responsibilities at each stage. Maintain it as systems and teams change.	Critical	Low	Cl.5.3; A.3.2
SG-2.03 **Clear allocation of responsibilities (external)**	Clarify obligations with external parties via contracts (security, performance, data protection); address third-party opacity.	Add AI-specific clauses to third-party contracts covering security arrangements, data handling, disclosures and audit rights. Evaluate vendor security features, and scope down or in-source where they are lacking.	High	Medium	A.10.2; A.10.3
SG-2.04 **Adaptive governance capability**	Build capability to understand fast-evolving agentic developments and update governance accordingly.	Assign a team to track agentic developments and schedule periodic governance reviews. Refresh policies, training and controls as new capabilities and risks emerge.	Medium	Medium	Cl.7.2; Cl.10
SG-2.05 **Define significant checkpoints requiring human approval**	Require human approval before high-stakes, irreversible, outlier, or user-defined actions.	Classify actions by risk and set system-enforced approval gates for high-stakes, irreversible, outlier and user-defined actions. Allow users to add their own thresholds, and log every approval decision.	Critical	Medium	A.9.2; A.6.2.6
SG-2.06 **Design effective approval requests**	Make requests contextual/digestible with risk clear; match input form (approve/reject, edit-plan, written justification).	Design concise approval prompts that show the action, its risk and a confidence score rather than raw logs. Enable plan editing for complex cases, and require written justification for high-risk approvals.	Medium	Medium	A.9.2; A.8.2
SG-2.07 **Train & audit human approvers**	Train humans to evaluate approvals effectively and audit those approvals.	Provide role-based training for approvers, and audit a sample of approvals periodically to confirm genuine scrutiny. Feed findings back into training.	High	Low	Cl.7.2; Cl.9.2
SG-2.08 **Guard against automation bias & alert fatigue**	Monitor override rates and response times; complement approvals with automated monitoring and alert thresholds.	Instrument approvals to track latency and override frequency. Flag reviewers who never override or approve too fast, rotate reviewers, and tune alert thresholds to avoid fatigue.	High	Medium	A.9.2; A.6.2.6
Dimension 3 - Implement Technical Controls & Processes					
SG-3.01 **Select control types deliberately**	Choose structural / rule-based / model-based / runtime controls appropriate to each risk; prefer system-level over prompt-layer.	For each risk, deliberately select the most robust control type, preferring structural and rule-based controls over prompt-layer ones. Use model-based safeguards for fuzzy risks and runtime controls for real-time interception.	High	Medium	A.6.2.3; A.8.4
SG-3.02 **Guardrails on planning & reasoning**	Reflect-on-plan, summarise-and-clarify, and log plan/reasoning for human verification.	Add plan-reflection and clarification steps before execution, and log the plan and reasoning trace for human verification. Require plan approval for high-risk workflows.	High	Medium	A.6.2.2; A.6.2.4

Control	Description	Implementation Strategy	Priority	Effort	ISO 42001 ref
SG-3.03 **Guardrails on tools**	Strict input formats; least-privilege tools via authN/authZ; no write to sensitive tables unless required; user takeover for secrets.	Define per-tool input schemas and permission scopes, default tools to read-only, and require user takeover for secret entry. Prohibit writes to sensitive tables unless essential, enforced through authorisation.	Critical	Medium	A.6.2.3; A.8.2
SG-3.04 **Protocol & MCP controls**	Use standard protocols; whitelist MCP servers; sandbox code; use MCP as a governance layer (filter/log/whitelist).	Maintain an MCP server whitelist and block others, sandbox any agent-executed code, and use the MCP layer to filter sensitive data and log all agent-to-system calls.	High	High	A.8.3; A.10.3
SG-3.05 **Multi-agent interaction controls**	Require structured schemas (typed calls) over free text; limit shared memory between agents.	Enforce typed, schema-based messages between agents rather than free text, scope shared memory to only what is necessary, and monitor agent-to-agent exchanges for anomalies.	Medium	High	A.6.2.3; A.6.2.6
SG-3.06 **Access controls, guardrails & approvals as core components**	Treat access controls, guardrails, and human approvals as first-class agent components limiting action-space/autonomy.	Bake access controls, guardrails and human approvals into the agent architecture from the start, and verify each is present and tested for every deployed agent.	Critical	Medium	A.6.2.3
SG-3.07 **Pre-deployment testing for baseline safety & reliability**	Test with realistic data/environments; verify controls and HITL work (attempt disallowed actions).	Write a test plan per agent covering functional, safety and control-verification tests using realistic and adversarial data. Attempt disallowed actions to confirm restrictions and human-in-the-loop trigger correctly before release.	Critical	Medium	A.6.2.4; Cl.8
SG-3.08 **Red-teaming & attack-vector testing**	Red-team use-case-specific vectors (e.g. indirect prompt injection) to validate guardrails.	Run red-team exercises against realistic attack scenarios including indirect prompt injection, before launch and periodically after. Track findings to closure and re-test.	High	High	A.6.2.4
SG-3.09 **Human-in-the-loop during testing**	Include human review during testing of high-autonomy agents; log reasoning per step to surface emergent behaviour.	Have reviewers observe test runs of high-autonomy agents and capture step-by-step reasoning. Feed emergent behaviours back into controls and datasets before launch.	High	Medium	A.9.2; A.6.2.4
SG-3.10 **Gradual / phased deployment**	Roll out gradually by users, tools/protocols, and systems to contain blast radius.	Define rollout phases with entry and exit criteria. Start with trained users, limited or no MCP access and low-risk systems, and expand only after monitoring confirms stability.	High	Medium	Cl.8.1; A.6.2.5
SG-3.11 **Continuous monitoring & multi-layer logging**	Monitor/log across layers with alert thresholds and anomaly detection; prioritise high-risk actions.	Log at every layer (user-agent, agent-tool, model reasoning), define programmatic and anomaly-based alerts with graded interventions, integrate observability tracing, and prioritise monitoring of high-risk actions.	Critical	High	A.6.2.6; A.6.2.8; Cl.9.1
SG-3.12 **Real-time intervention & failsafe mechanisms**	Stop-and-escalate on failure; termination/fallback for catastrophic malfunction or compromise.	Implement a tested stop or kill control per agent, define escalation paths and on-call ownership, predefine safe fallback states, and rehearse recovery.	Critical	Medium	A.6.2.6; A.9.2
SG-3.13 **Log immutability & audit trails**	Maintain tamper-proof audit trails including blocked actions and failures.	Store logs in append-only or tamper-evident storage such as write-once media or hash-chaining, retain per policy, and never allow deletion of problematic agent trajectories.	High	Medium	A.6.2.8; A.4.6
SG-3.14 **Regular audit of agent performance**	Audit at intervals to confirm expected performance and detect drift.	Schedule periodic performance and control audits, test for drift, and track findings to remediation with reporting to the governance committee.	Medium	Low	Cl.9.2; Cl.9.1
SG-3.15 **Robust change management**	Define change triggers, categorise changes by risk, and re-risk critical changes so small changes don't cascade.	Route all changes (model, tools, prompts, autonomy, thresholds) through change control. Apply light review for prompt tweaks and full governance review plus re-risk for model or autonomy changes.	High	Medium	Cl.8.1; Cl.6.3

Control	Description	Implementation Strategy	Priority	Effort	ISO 42001 ref
SG-3.16 **Structural separation of sensitive data**	Keep sensitive data out of agent context (e.g. TEE + opaque references) so it cannot be leaked/extracted.	Hold sensitive data in a secure boundary such as a trusted execution environment, and give the agent only opaque references. Verify no sensitive data enters the agent's prompt or context.	High	High	A.8.2; A.7.2
Dimension 4 - Enable End-User Responsibility					
SG-4.01 **Tailor enablement to different user types**	Recognise interacting users vs integrating users have different information/oversight needs.	Segment users into those who interact with agents and those who integrate them, and design distinct guidance for each. Give integrating users deeper oversight tooling.	Medium	Low	A.9.3; A.8.2
SG-4.02 **Equip users to use agents appropriately**	Provide essential information to use agents correctly and oversee them.	Publish clear usage guidance covering capabilities, limitations, do and do-not lists and escalation routes, and surface it at first use.	High	Low	A.8.2; A.9.2
SG-4.03 **Enable oversight by integrating users**	Support users embedding agents into workflows; manage model changes centrally, not via end users.	Give integrating users monitoring visibility and controls, and handle model and tool updates through managed change processes rather than ad hoc end-user changes.	Medium	Medium	A.9.2; A.6.2.5
SG-4.04 **Preserve tradecraft & business continuity**	Prevent skill erosion from over-reliance; ensure continuity if agents fail/unavailable.	Identify critical skills at risk of atrophy, maintain manual fallback procedures and periodic no-agent drills, and include agent outage in business-continuity planning.	Medium	Medium	Cl.7.2
SG-4.05 **Personnel training on autonomous-agent risks**	Train personnel on agent risks and reinforce responsibility to prevent careless misuse.	Run recurring training on autonomous-agent risks and responsible use, track completion, and refresh on major capability or policy change.	High	Low	Cl.7.2; Cl.7.3
Cross-Cutting - Iterative Governance					
SG-X.01 **Iterative re-evaluation across dimensions**	Anomalies during implementation/monitoring trigger re-evaluation of earlier dimensions; continuous improvement.	Define feedback loops from monitoring and incidents back into risk assessment and design. On any anomaly, re-run the relevant dimension's controls and tighten bounds before continuing.	High	Medium	Cl.10.1; Cl.9.1

Priority and Effort Summary

By priority (51 controls)

Priority	Count	Share
Critical	15	29.4%
High	27	52.9%
Medium	9	17.6%
Total	**51**	**100%**

By effort

Effort	Count	Share
High	9	17.6%
Medium	28	54.9%
Low	14	27.5%
Total	**51**	**100%**

Roadmap tip: the Critical and Low-effort controls are your fastest risk reduction. Tackle those first, then the Critical and Medium or High-effort controls, before moving through High and Medium priorities.

Source

- Singapore Model AI Governance Framework for Agentic AI, v1.5 (IMDA and PDPC, published 20 May 2026, updated 5 June 2026), from project files.
- ISO/IEC 42001:2023 structure, used for alignment references, paraphrased not quoted.

Follow and Subscribe

This checklist is part of the CXO Intelligence Series on Governance by Prashant Akhawat. If you found it useful, connect and subscribe for more practical AI governance resources.

Website: akhawat.com

LinkedIn: linkedin.com/in/prashantakhawat

CXO Intelligence Series: Subscribe on LinkedIn