ANOVA CLOUD

Agentic AI Readiness Assessment

A structured assessment for deciding whether your use cases, data, tools, evals, security, and operating model are ready for production agents.

Discovery questionnaire · Working document
Complete with your AnovaCloud architect during the discovery engagement.
Bring evidence, not opinions — every rating should point to an artifact, owner, or system.

How to use this questionnaire

- Work through this with the people who own the answers — architects, data engineers, security, and business sponsors. One respondent per section rarely knows enough.
- Answer for **today's reality**, not the roadmap. Roadmaps go in the notes, clearly labeled.
- Where a question doesn't apply, mark N/A and say why — a wrong answer is worse than a blank one.
- Your AnovaCloud architect will challenge ratings in the workshop. Evidence wins arguments.

Rate each statement 1–5 where a scale is shown. **1** = not true / absent. **5** = fully true / mature and evidenced. Use the notes lines to cite the artifact, system, or person behind the rating — ratings without evidence are revisited in the workshop.

A Use-case candidacy

Not every workflow deserves an agent. Score candidates honestly.

A.1 Candidate workflows decompose into distinct steps with different tools or skills.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____

_____

A.2 Task success can be defined and verified without trusting the agent's word.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____

_____

A.3 The cost of an agent being wrong is understood and acceptable (or mitigated by design).

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____

_____

A.4 Workflows run at a volume where automation pays for agent operating cost.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____

_____

A.5 There are tasks where a simpler approach (RAG, script, RPA) has been ruled out for cause.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____

_____

A.6 Stakeholders accept probabilistic systems — 'usually right, verifiably checked' over 'deterministic'.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____

_____

A.7 At least one candidate has an owner willing to co-design and evaluate with the build team.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____

_____

B Data & API surface

Agents are only as capable as the systems they can touch.

B.1 The systems agents must read (docs, tickets, databases) have accessible, permissioned APIs.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____

_____

B.2 The systems agents must write to (CRM, ITSM, deployment, comms) have safe, scoped APIs.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____

_____

B.3 API rate limits, costs, and failure modes are documented for agent-scale calling patterns.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____

_____

B.4 Test/sandbox environments exist for every system the agent will touch.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____

_____

B.5 Data the agent needs is current and trustworthy — agents amplify bad data faster than dashboards do.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____

_____

B.6 Retrieval over private knowledge (RAG) is available or can be built as the agent's memory.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____

_____

B.7 Secrets and credentials for integrations are managed centrally (vault), not in prompts or code.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____

_____

C Tool & integration readiness

Tools are the agent's hands — design them like an API contract.

C.1 Each tool an agent needs can be defined with a clear contract: inputs, outputs, errors, side effects.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____

_____

C.2 Destructive or externally visible actions are identifiable and can be gated.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____

_____

C.3 Tool allowlists per agent role are feasible (least privilege, not one master key).

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____

_____

C.4 Long-running tools (provisioning, batch jobs) support async patterns (poll, callback).

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____

_____

C.5 Tool versioning and change management exist — agents break silently when tools change.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____

_____

C.6 Idempotency is understood: retried tool calls must not double-charge, double-send, or double-provision.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____

_____

D Evaluation & observability

If you can't evaluate it, you can't ship it.

D.1 Golden task sets exist or can be built: realistic tasks with graded outcomes.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____

_____

D.2 Task-level success criteria are defined per candidate workflow.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____

_____

D.3 Someone owns eval maintenance — golden sets rot as products change.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____

_____

D.4 Tracing infrastructure can capture agent plans, tool calls, and intermediate states.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____

_____

D.5 Cost observability per run/task is available or planned (tokens × volume gets expensive fast).

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____

_____

D.6 Production feedback loops exist (thumbs, corrections, escalations) feeding back into evals.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____

_____

D.7 There is a defined bar for 'good enough to expand autonomy' — not a vibe.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____

_____

E Security & policy

Answer with security and legal in the room.

E.1 Threat model for agents exists: prompt injection via tool outputs, data exfiltration, rogue tool use.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____

_____

E.2 Policy gates can be enforced deterministically for cost, risk, and compliance rules.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____

_____

E.3 Data classification is applied to what agents can read — agents must not widen access.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____

_____

E.4 PII handling rules for agent inputs, tool calls, and logs are defined.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____

_____

E.5 Incident response covers agent misbehavior (wrong actions at machine speed).

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____

_____

E.6 Vendor/model contracts cover agentic use: data retention, tool-call logging, liability.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____

_____

E.7 Red-teaming of agent systems is planned before production expansion.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____

_____

F Human oversight & change

Autonomy is earned, and humans stay in the loop where it matters.

F.1 High-stakes actions are enumerated and mapped to approval requirements.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____

_____

F.2 Approvers have the context to decide well — not just an 'approve' button on a black box.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____

_____

F.3 Kill switches (per-run and global) are technically feasible in the target architecture.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____

_____

F.4 End users affected by agent outputs are identified and involved in design.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____

_____

F.5 Escalation paths are defined: when the agent is uncertain, a human gets a good handoff.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____
_____

F.6 The organization has appetite for iterative autonomy — starting supervised, expanding on evidence.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 *(1 = not true / absent → 5 = fully true / mature)*

Evidence / notes: _____
_____

Maturity model

Average all ratings (ignore N/A) for an overall maturity read — then average per section to find your constraint. Overall maturity sets ambition; the weakest section sets the plan.

Maturity	Signal (avg)	What it means
Level 1 · Exploring	1.0 – 2.0	Agent-shaped ideas, no foundation. Start with RAG or automation; revisit agents after data/API gaps close.
Level 2 · Preparing	2.1 – 2.9	Some candidates viable. Run a supervised pilot on one workflow with tight evals and human approval on all writes.
Level 3 · Piloting	3.0 – 3.6	Ready for a production pilot: scoped tools, policy gates, full tracing, golden evals gating every change.
Level 4 · Operating	3.7 – 4.4	Expand to adjacent workflows. Tier autonomy by risk; invest in eval operations and red-teaming as standing functions.
Level 5 · Scaling	4.5 – 5.0	Agent platform territory: shared tool registry, policy-as-code, org-wide eval standards. Rare — verify before claiming.

Next steps

- Average each section; the lowest-scoring section is your constraint — programs fail at their weakest dimension, not their strongest.
- Typical next step: an **Agent Deployment Accelerator** — one workflow from candidate to supervised production with evals, gates, and tracing.
- Bring this completed assessment to your discovery call. **anovacloud.com/contact/**

AnovaCloud — Engineering the AI-Native Enterprise. Advisory · Engineering · Agentic AI · Modernization · Academy · Workforce. Frisco, Texas. **Talk to an Architect:** anovacloud.com/contact/