Orquestra AI

AGENTS DRIVE · HUMANS NAVIGATE

The Assessment Guide

Orquestra Navigator™ · How Governable Autonomy Is Measured

The Framework document establishes what Governable Autonomy is.
This document establishes how it is measured.

DISTRIBUTION: NOT YET RELEASED · INTERNAL REVIEW DRAFT

USE: NAVIGATOR ASSESSORS · COMMISSIONING LEADERSHIP

PREFACE

How to Read This Document

Per the Navigator V1.0 design charter, content in this guide is labeled STABLE or PROVISIONAL. Stable content, the Capability Model's structure, the diagnostic pipeline's shape, the finding taxonomy, is unlikely to change materially and can be relied on now. Provisional content, specific thresholds, weighting, the archetype applicability table's edge cases, is a working hypothesis pending pilot validation against real organizations. Both are marked inline throughout.

Treating a Provisional figure as settled is the overclaiming failure this framework exists to avoid. Please don't.

I

What This Guide Adds to the Framework

The Framework whitepaper establishes that Navigator measures one thing, Governable Autonomy, using five capabilities as evidence. It does not specify how a capability is scored, how a system's applicable capabilities are determined, or how the five combine into a single defensible number. That is the entire subject of this document.

> STABLE
>
> **The output of an assessment is one of two things: a Certified Stage, evidenced by an assessor against gate criteria, or an Indicative Stage, a self-reported estimate explicitly labeled as unverified.**

Everything in this guide exists to make that number, whichever it is, honest.

II

The Navigator Capability Model, in Full

STABLE

Each capability that follows uses a consistent nine-part structure, so that assessments are repeatable and an assessor's judgment is checked against the same criteria every time, regardless of who runs the engagement.

1. **Purpose.** Why this capability exists and why it is essential to Governable Autonomy.
2. **Why It Matters.** How it contributes to safe delegation, and the risk when it is weak.
3. **Capability Evolution (Levels 0-4).** How it progresses from traditional delivery to continuous assurance.
4. **Required Skills.** The competencies needed to operate safely at each level.
5. **Engineering Artifacts.** Observable implementation evidence.
6. **Observable Evidence.** What an assessor actually looks at during assessment.
7. **Assessment Criteria.** What must be true, in evidence, to satisfy a given level.
8. **Common Anti-Patterns.** Typical weaknesses that block safe autonomy.
9. **Relationships to Other Capabilities.** How weakness here constrains, or is constrained by, the other four.

The five capabilities that follow, in the order assessed: Human Responsibility, Decision Governance, Engineering Capability, Knowledge Capability, and Assurance Capability.

CAPABILITY 1

Human Responsibility

1. PURPOSE

To ensure that as decision-making authority moves to AI, a human retains and can discharge responsibility for the outcome, at every level of autonomy.

2. WHY IT MATTERS

Autonomy without a responsible human is not governance, it is abdication. This capability keeps "who is accountable" answerable at every stage, including stages where no one reviews individual outputs anymore.

3. CAPABILITY EVOLUTION

Level 0. A human executes the work directly; responsibility is total and undivided.
Level 1. A human reviews every AI output before it advances; responsibility is discharged through review.
Level 2. A human supervises bounded AI-executed tasks at defined checkpoints; responsibility is discharged through checkpoint judgment on cumulative output, not line-by-line inspection.
Level 3. A human governs the boundaries AI operates within and is authoritative at escalation; responsibility is discharged through escalation-architecture design and competent resolution when called.
Level 4 (forward-looking). A human assures, at a distance, that approved objectives are being met; responsibility is discharged through objective-setting and statistically rigorous periodic review.

4. REQUIRED SKILLS

STABLE CONCEPT, PROVISIONAL DETAIL

Levels 0-2: domain and software engineering competence, critical evaluation of AI output, checkpoint judgment. Level 3: escalation design, authority to decide under partial context and time pressure, resistance to auto-approval drift. Level 4: statistical sampling literacy, drift interpretation, comfort holding accountability for outcomes observed only in aggregate. Maps directly to the Navigator Evaluator's staged profiles: **Agent Director** (Levels 1-2), **Loop/Harness Architect** and **Resolver** (Level 3), **Audit Owner** (Level 4, not yet a fielded instrument).

5. ENGINEERING ARTIFACTS
Named role assignments (harness owner, resolver, objective owner). Escalation-response training records. Fatigue-metric dashboards.

6. OBSERVABLE EVIDENCE
Who actually approved a sampled set of past decisions, and whether they can explain the basis without re-running the AI. Resolver response times and auto-approve rates. Whether a named owner exists for the system in scope.

7. ASSESSMENT CRITERIA
PROVISIONAL

A system claiming Level 3 must show, for every escalation trigger class, a named resolver with documented authority and a sampled record of context-grounded decisions. Absence of a named resolver for any trigger class caps this capability at Level 2 for that system, regardless of technical sophistication elsewhere.

8. COMMON ANTI-PATTERNS
A resolver who approves without reading provided context. A "named owner" who is a title with no real engagement. Checkpoint reviewers rubber-stamping because volume outpaced real capacity to evaluate. Treating Level 3 resolver assignment as a formality rather than a competency-gated role.

9. RELATIONSHIPS TO OTHER CAPABILITIES
Most directly threatened by weakness in Assurance Capability: without fatigue metrics and near-miss review, degraded Human Responsibility (auto-approval creep) is invisible until it produces an incident. Also the capability the Navigator Evaluator exists to assess directly, making it the most likely to be scored via a dedicated hiring or promotion instrument rather than solely through organizational assessment.

CAPABILITY 2

Decision Governance

1. PURPOSE

To ensure the organization can define, delegate, constrain, and revoke AI decision-making authority, deliberately rather than by default.

2. WHY IT MATTERS

Without designed delegation, authority accrues to AI systems by accident, through the accumulation of unreviewed exceptions and undocumented workarounds, not through anyone's decision. This capability makes the boundary of AI authority a designed artifact rather than an emergent one.

3. CAPABILITY EVOLUTION

Level 0. No delegation exists; all decisions are human-made.
Level 1. An AI-usage policy exists, stating where AI may act.
Level 2. Loop-scoping practice exists: task boundaries, permitted actions, and checkpoint placement documented per workflow.
Level 3. A governance-approved escalation trigger catalogue exists, mapping specific risk and authority thresholds to specific resolver roles, under formal change control.
Level 4 (forward-looking). Upstream objective and boundary approval precedes every loop's operation; periodic re-validation of risk classification is scheduled and enforced.

4. REQUIRED SKILLS

Policy design, escalation trigger design, and enough understanding of the risk-ceiling determinants (Framework, Section VI) to translate them into concrete, testable triggers rather than vague guidance.

5. ENGINEERING ARTIFACTS
The AI-usage policy document. The loop-scoping template. The escalation trigger catalogue itself, version-controlled. Change-control records for the catalogue.

6. OBSERVABLE EVIDENCE
Whether the trigger catalogue is actually referenced in incident postmortems. Whether a sampled set of AI-executed workflows can be traced back to a documented scope. Whether catalogue changes show a governance sign-off trail.

7. ASSESSMENT CRITERIA
PROVISIONAL

Level 3 requires the catalogue to cover every distinct escalation pathway in active use for the system in scope, each mapped to a named resolver role (Capability 1) and a specific risk/authority threshold, with evidence the catalogue has been adversarially tested for gaps (see Capability 5).

8. COMMON ANTI-PATTERNS
A policy that exists but is never consulted. A trigger catalogue written once at the Level 2 to 3 transition and never revisited as system behavior changes. Scope creep where a loop originally bounded to a narrow task quietly expands without a corresponding governance update.

9. RELATIONSHIPS TO OTHER CAPABILITIES
Without Engineering Capability, Decision Governance is a policy with nothing enforcing it: a trigger catalogue that exists only on paper, not as a hard stop in the harness, is not a control. Without Assurance Capability, it cannot detect its own gaps; the two must be read together.

CAPABILITY 3

Engineering Capability

1. PURPOSE

To ensure the organization can build the technical harness that makes a given level of autonomy real rather than aspirational.

2. WHY IT MATTERS

Governance design with no engineering counterpart is fiction. A documented escalation trigger the system cannot actually enforce as a hard stop provides no protection at all; it provides the appearance of protection, which is worse.

3. CAPABILITY EVOLUTION

Level 0. Standard software engineering; no AI-specific harness exists.
Level 1. AI-assisted tooling integrated with provenance tagging of generated output.
Level 2. A real harness exists: sandboxed least-privilege execution, technically enforced checkpoints, per-iteration logging sufficient for reconstruction, a demonstrated kill switch.
Level 3. Escalation triggers enforced as hard stops, verified by adversarial test. Full traceability, including triggers evaluated but not fired. Automated rollback for reversible action classes.
Level 4 (forward-looking). Continuous telemetry with drift detection against validated baselines. Automated stage-downgrade: the harness re-imposes Level 3 behavior on defined signals, without waiting for a human decision.

4. REQUIRED SKILLS

Sandbox and permission architecture, harness pipeline engineering, adversarial testing design, telemetry and observability engineering at Level 4.

5. ENGINEERING ARTIFACTS
The harness codebase itself. Sandbox configuration. Trigger enforcement code. Logging and telemetry infrastructure. Kill-switch implementation and its test records.

6. OBSERVABLE EVIDENCE
A live demonstration that an out-of-bounds action is actually blocked, not merely logged. A live demonstration that the kill switch works under load. Logs that actually reconstruct a sampled loop's behavior end to end.

7. ASSESSMENT CRITERIA
PROVISIONAL

Level 3 certification requires a passed adversarial evasion test performed within the current recertification cycle, not merely at initial build. A mechanism not re-tested since a significant harness change does not count as current evidence.

8. COMMON ANTI-PATTERNS
A kill switch that exists in code but has never actually been triggered in a realistic drill. Logging that captures what happened but not what was evaluated and didn't fire, silently weakening the system's only partial defense against the silent-failure problem. Treating a one-time sandbox test as permanent evidence.

9. RELATIONSHIPS TO OTHER CAPABILITIES
The substrate Decision Governance's designs run on and Assurance Capability's evidence is drawn from. Weakness here undermines both regardless of how well-designed they are on paper.

CAPABILITY 4

Knowledge Capability

1. PURPOSE

To ensure that where an AI system reasons from organizational knowledge, that knowledge is governed, current, and distinguishable from the system's own behavioral drift.

2. WHY IT MATTERS

A system can have excellent execution controls and still make confidently wrong decisions if what it reasons from is stale, unsourced, or ungoverned. This is an independent failure mode: no amount of escalation design or sandboxing catches a decision that was correct given the system's inputs but wrong because the inputs themselves were bad.

3. CAPABILITY EVOLUTION

Level 0-1. Knowledge sources, if any, are informal and unversioned.
Level 2. Basic source control exists for policies and business rules the system draws on.
Level 3. Retrieval is policy-governed; retrieval quality is actively monitored.
Level 4 (forward-looking). Knowledge drift detection runs integrated with the system's behavioral drift detection, so staleness in what the system knows is distinguished from drift in how it behaves.

4. REQUIRED SKILLS

Knowledge governance, retrieval system design, and the judgment to distinguish a knowledge problem from a control problem when a decision goes wrong.

5. ENGINEERING ARTIFACTS
Versioned policy and business-rule stores. Retrieval pipeline configuration. Source-attribution logging.

6. OBSERVABLE EVIDENCE
Whether a sampled AI decision's stated basis can be traced to a current, correctly sourced policy document. Whether retrieval quality is measured at all.

7. ASSESSMENT CRITERIA
PROVISIONAL

Applicability-gated (Section III): scored only where the system in scope actually depends on retrieved organizational context. Where Required, Level 3 requires evidence of active retrieval quality monitoring, not merely that a retrieval system exists.

8. COMMON ANTI-PATTERNS
Treating a RAG pipeline's mere existence as evidence of governance. No process for retiring or updating stale source documents. Attributing a bad decision entirely to "the model" when the actual cause was an outdated policy document it correctly retrieved.

9. RELATIONSHIPS TO OTHER CAPABILITIES
Uniquely among the five, not universally load-bearing: its relevance depends on system archetype (Section III). Where relevant, interacts closely with Assurance Capability, since knowledge drift and behavioral drift require related but distinct monitoring.

CAPABILITY 5

Assurance Capability

1. PURPOSE

To ensure the organization can continuously demonstrate, with evidence, that the other four capabilities are actually operating as designed, not merely that they were designed well.

2. WHY IT MATTERS

This is the capability that turns a design into a fact. Decision Governance answers whether a control was well designed; Assurance Capability answers whether it is actually working, a distinct question an organization can get badly wrong in either direction.

3. CAPABILITY EVOLUTION

Level 0-1. No structured monitoring beyond incident response.
Level 2. Incidents are triaged and root-caused; a genuine incident procedure exists.
Level 3. Scheduled near-miss and blocked-escalation review exists, alongside adversarial trigger-evasion testing performed on a defined cadence.
Level 4 (forward-looking). Drift-response procedures with automatic reversion thresholds exist; statistically defensible sampling design underlies periodic audit.

4. REQUIRED SKILLS

Incident analysis, adversarial testing design; at Level 4, statistical sampling design and drift interpretation, a skill profile the market currently has little of.

5. ENGINEERING ARTIFACTS
Near-miss review records. Evasion-test reports and their remediation trail. Fatigue-metric dashboards (shared with Capability 1). Drift-detection configuration and its alert history.

6. OBSERVABLE EVIDENCE
Whether near-miss reviews actually happened on the stated cadence, with real findings, not a rubber-stamped log. Whether an evasion test has ever actually found something, and what happened after it did.

7. ASSESSMENT CRITERIA
PROVISIONAL

Level 3 requires at least one completed adversarial evasion test cycle with a documented remediation outcome within the current period, not merely a scheduled intention to test.

8. COMMON ANTI-PATTERNS
Monitoring dashboards nobody looks at. A near-miss review meeting that always concludes "nothing to report." Evasion testing performed once at initial certification and never repeated. Treating the mere existence of a monitoring tool as evidence the organization is watching it.

9. RELATIONSHIPS TO OTHER CAPABILITIES
The check on every other capability's claims. Also the one whose own weakness is hardest to self-detect, since an organization poor at assurance is, by definition, poor at noticing that it's poor at assurance. External assessment matters more here than elsewhere in the model.

III

Capability Applicability

STABLE MECHANISM PROVISIONAL TABLE

Not every capability constrains every system equally. Applicability is determined primarily by **system archetype**, not by assessor discretion, specifically to prevent an assessor or an organization from marking an inconvenient capability "Not Applicable" to inflate a result.

ARCHETYPE	HUMAN RESP.	DECISION GOV.	ENGINEERING	KNOWLEDGE	ASSURANCE
Pure code-gen / dev loop	Required	Required	Required	N/A (Relevant at most)	Required
RAG-based decisioning agent	Required	Required	Required	**Required**	Required
Deterministic classifier (no retrieval)	Required	Required	Required	N/A	Required
Customer-facing conversational agent	Required	Required	Required	Required	Required
Internal ops automation (low blast radius)	Required	Required	Relevant	Relevant	Relevant
Other / custom	Required	Required	Required	Assessor-declared, justification required	Required

Rules. Required capabilities participate in the Capable Stage min-rule. Relevant capabilities are scored and inform recommendations but do not cap Capable Stage. Not-Applicable capabilities are excluded entirely. Overriding a default requires a recorded, mandatory justification; an override without one is rejected, not silently accepted. Human Responsibility and Decision Governance are never Not-Applicable for any archetype: every system has an accountable human and a decision-authority structure, even a minimal one.

IV

The Assessment Pipeline

STABLE STRUCTURE

Navigator Capability Model

↓

Capability Assessment (score each Required/Relevant capability, 0–4)

↓

Capable Stage = min(score of each Required capability)

↓

Risk Ceiling = min(R1..R5), assessed per system

↓

Maximum Permitted Stage = min(Capable Stage, Risk Ceiling)

↓

Actual Operating Stage (self-reported or observed)

↓

Evidence Assessment (does evidence support the claimed stage?)

↓

Certified Stage (Section V)

↓

Diagnostic Finding (Section VI)

V

Certified Stage: The Formula

STABLE

Resolves the Navigator V1.0 design charter's Section 6.1 open item.

```
Capable Stage           = min(score of each Required capability)
Risk Ceiling            = min(R1..R5)
Actual Stage            = self-reported/observed operating mode
Evidence-Supported Stage  = highest stage N such that Gate (N-1)->N evidence
                    requirements are confirmed, checked sequentially
                    from Stage 0 upward, stopping at first failure.
                    = "Unverified" if this is a self-serve pass with
                    no assessor-led evidence review.

Certified Stage = min(Actual Stage, Capable Stage, Risk Ceiling, Evidence-Supported Stage)

 -- if Evidence-Supported Stage = "Unverified":
   Certified Stage is NOT computed.
   Report "Indicative Stage" = min(Actual, Capable, Risk Ceiling) instead,
   labeled explicitly as unverified and not a certification.
```

This is the rule that makes two assessors reach the same conclusion from the same evidence, and the rule that prevents a self-serve pass from quietly producing a number that reads as audited when it isn't. No version of this guide, or any tool built from it, should compute a Certified Stage without a real Evidence-Supported Stage behind it.

VI

Diagnostic Findings

STABLE

An assessment does not terminate in a score. It terminates in a diagnosis, comparing Actual Stage against the other three measures:

FINDING	CONDITION	WHAT IT MEANS
Governance Failure	Actual > Risk Ceiling	Operating beyond what is permitted for this system. The most urgent finding: reduce autonomy or re-validate the risk classification with evidence.
Assurance Failure	Actual > Evidence-Supported Stage	Claiming controls that cannot be demonstrated. Often the most dangerous finding, because it looks fine from the outside.
Competitiveness Finding	Actual < min(Capable, Risk Ceiling)	Unnecessary overhead relative to what is both possible and permitted. Advisory, not compliance.
Aligned	Actual = Certified Stage	The organization is operating exactly where its evidence and its ceiling say it should.

Example output

```
Capable Stage:    4
Risk Ceiling:     3
Actual Stage:     4
Certified Stage:  2
-> Finding: GOVERNANCE FAILURE
```

VII

The Autonomy Contract Template

STABLE

Every assessment produces or updates one of these, per system. It is not an appendix; it is the operational artifact the rest of the framework refers back to.

AUTONOMY CONTRACT

System name: _____
Business purpose: _____
System owner (named): _____
Archetype: _____

Risk classification:
Reversibility: ___ (Full / Effortful / Limited / Irreversible)
Named accountability: ___ (None / Aggregate / Per-decision)
Explainability obligation: ___ (None / Internal / Legal)
Regulatory classification: ___
Blast radius: ___ (Single artifact / Team-product / Systemic)
--> Risk Ceiling: Level ___

Capability scores (Required only, this system):
Human Responsibility: Level ___
Decision Governance: Level ___
Engineering Capability: Level ___
Knowledge Capability: Level ___ (or N/A, archetype default / override + justification)
Assurance Capability: Level ___
--> Capable Stage: Level ___

Actual Operating Stage: Level ___ (self-reported / observed)
Evidence-Supported Stage: Level ___ (or Unverified)
--> Certified Stage / Indicative Stage: Level ___

Diagnostic Finding: _____

Authority boundaries: _____
Escalation rules & resolver(s): _____
Evidence requirements: _____
Kill switch owner & test date: _____
Review / recertification cadence: _____

Last updated: _____
Next recertification due: _____

VIII

What This Guide Does Not Yet Resolve

Consistent with the design charter and the Framework whitepaper's own discipline, named rather than hidden:

- Fatigue-metric and near-miss cadence thresholds are asserted, not empirically derived.
- The archetype applicability table's "Other/custom" row needs real-engagement calibration.
- Assessment Criteria marked Provisional throughout Section II are working hypotheses; none have been field-tested.
- The Certified Stage formula assumes an honest, competent gate review behind Evidence-Supported Stage. It inherits whatever rigor that review has and cannot detect a superficial one.
- This guide does not and cannot resolve the silent-escalation-failure problem described in the Framework whitepaper, Section X. No instrument in this family will.

Where to Go Next

Just need the concepts? Start with the Orquestra Navigator™ Framework whitepaper.

Implementing controls? Start with the Navigator Controls Catalogue and the Critical Controls Guide for your system's risk classification.

Hiring for AI-native engineering roles? Start with the Navigator Evaluator.

Running an assessment right now? This document, plus the Autonomy Contract template in Section VII, is what you need in hand.

This document establishes how Governable Autonomy is measured. What it measures is defined in the Framework.

Rajesh Srinivasan
Orquestra AI · rajesh@orquestraai.io
orquestraai.io

Orquestra Navigator™ Assessment Guide · Working draft, not yet released.