LLM INQUISITOR – Practitioners Guide

SUBJECT YOUR A.I. TO THE FULL LLM INQUISITION

COPYRIGHT NOTICE

LLM INQUISITOR - Quick Start Guide V 1.0

1. Introduction: What This Guide Is For

1.1 AI doesn't fail in labs - it fails in workflows.

It fails in email threads, document edits, coding sessions, analysis tasks, and customer-facing interactions. That's where drift, collapse, contradiction, and instability actually matter.

1.2 LLM INQUISITOR exists to reveal that behaviour.

This guide gives practitioners a simple, reliable way to evaluate an AI the way it is actually used: normal tasks, normal expectations, normal pressure. No puzzles. No adversarial prompts. No artificial test harnesses. Just real work.

When you use an AI normally, its behaviour becomes visible:

- how stable it is
- how it handles structure
- how it recovers
- how it drifts
- how it collapses
- how it behaves over time

This guide shows you how to capture that behaviour, classify it, and understand its impact without needing the full methodology or annex suite. The Practitioner's Guide is the operational entry point into INQUISITOR - the part you use every day.

LLM INQUISITOR METHODOLOGY vs Quick Start Guide:

Whereas the Quick Start Guide aimed to get you started in five minutes and the full Methodology is treated as the canonical reference designed for full operational evaluation, most people sit somewhere in the middle. This guide exists to bridge that gap.

It is for practitioners (developers, tester QA etc and users etc) who want to use INQUISITOR properly in real work, without needing to memorise annexes or run a full behavioural surface. It gives you the practical layer: how to think, how to observe, how to classify, and how to run a clean evaluation without drowning in detail.

INQUISITOR is not a prompt book. It is not a puzzle-solver. It is not a way to “beat” an AI. It is a way to understand how an AI behaves under real conditions - the same conditions you and your team work in every day.

When you use an AI normally, without tricks or adversarial intent, its behaviour reveals itself. This guide teaches you how to notice that behaviour, how to capture it, and how to describe it in a way others can act on.

You do not need to be an expert. You do not need special terminology. You do not need to run every test surface.

You only need to follow a simple workflow, observe honestly, and classify what you saw. The Practitioner’s Guide gives you the tools to do that with confidence.

The full methodology is there when you need depth, precision, and reproducible evidence. This guide is here to help you get there.

2. The INQUISITOR Mindset

LLM INQUISITOR is a form of operational testing. You are testing the system — but you are testing it under the same conditions it would normally be used for work. This is the behaviour that matters in real environments, not synthetic puzzles or adversarial traps.

The mindset is simple: use the AI the way it would normally be used for work and observe how it behaves. Real tasks reveal real behaviour. You don’t need tricks or stress-tests to expose weaknesses. They appear naturally when the system is placed inside a normal workflow.

INQUISITOR is not about breaking the model. It is about revealing its behaviour.

When you work the way you normally would, the system shows you:

- how stable it is
- how well it follows expectations
- how it handles structure and continuity
- how it recovers from mistakes
- how it behaves over time

This is the behaviour that determines whether an AI is safe and reliable in real workflows.

The INQUISITOR mindset is built on five principles:

- **Operational testing** - evaluate the model in the same conditions it will be used.
- **Work normally** - use real tasks, real instructions, real expectations.
- **Stay neutral** - don't rescue the model too early or steer it toward success.
- **Observe, don't hunt** - behaviour emerges naturally; you don't need adversarial prompts.
- **Failures are signals** - every wobble, drift, or collapse is data, not frustration.

This mindset ensures that INQUISITOR reflects real-world performance, not artificial test-bench behaviour. It keeps the method grounded, honest, and directly applicable to operational environments.

3. The Practitioner Workflow

The Practitioner Workflow is the expanded version of the Quick Start. It gives you enough structure to run a proper INQUISITOR session without needing the full methodology. Each step is simple, direct, and designed for real work.

You do not need special prompts.
You do not need to force failures.
You only need to follow the workflow and observe what happens.

3.1 Step 1 — Pick a Real Task

A real task is something the AI would normally be used for in your work. It should be practical, familiar, and connected to an actual workflow.

Good examples include:

- writing an email
- summarising a document
- fix a paragraph
- generate code
- analyse data
- rewrite a section of text
- produce a short explanation

Avoid synthetic or puzzle tasks. They distort behaviour and do not reflect real conditions.

A real task exposes real behaviour.

3.2 Step 2 — Declare Your Expectations

Expectations define the load envelope.
They tell the AI what success looks like and give you a baseline for judging behaviour.

One sentence is enough:

- "I expect it to keep the structure."
- "I expect it to follow my editing instructions."
- "I expect it to remember the variable names."
- "I expect it to critique only this document."

Clear expectations make classification easier later.

3.3 Step 3 — Work Normally

Use the AI the way it would normally be used for work.

Do not switch into test mode.

Do not try to break it.

Do not over-engineer prompts.

Normal usage reveals natural behaviour.

This is the behaviour that matters in real workflows.

3.4 Step 4 — Note When Something Feels Wrong

When something feels off, write a single line describing it.

This is your evidence anchor.

Examples:

- "It rewrote the whole thing despite instructions not to."
- "It forgot the earlier decision."
- "It merged two documents."
- "It changed the variable names without being told to."

Anchors are observations, not explanations. They capture what happened, not why.

3.5 Step 5 — Keep Going

Do not stop at the first failure.

INQUISITOR cares about the pattern, not the moment.

As you continue, observe:

- does it recover
- does it get worse
- does it drift

- does it collapse

This reveals the behavioural trajectory, which is often more important than the initial error.

3.6 Step 6 — Classify What You Saw

Use simple, practical labels:

- drift
- collapse
- contamination
- contradiction
- narrowing
- over-assertion
- loss of structure
- loss of persona
- loss of context
- hallucination

You do not need to be precise.

You only need to be directionally correct.

3.7 Step 7 — Decide the Severity

Severity is judged using three levels:

- **A** — stable
- **B** — unstable but recoverable
- **C** — collapse

If it breaks your workflow, it is a C.

A single C-event is enough to justify deeper evaluation.

This is the point where you decide whether to escalate to the full methodology.

---

4. Evidence

Evidence is the short note you make when the AI does something that matters for the task. It is a factual record of what the system actually did at a specific moment. Evidence is not about surprise or expectation. It is simply the behaviour you observed while working.

Evidence is one line. It describes what happened. It does not explain, justify, or diagnose.

Good evidence is clear and specific. Weak evidence is vague or emotional.

4.1 What Evidence Is

Evidence is a short, factual note describing a single behaviour that affected the task.

Examples:

- "It rewrote the entire section even though I asked for a small edit."
- "It forgot the variable names introduced earlier."
- "It added content from a different document."
- "It contradicted its earlier answer."

These are plain descriptions of what happened.

4.2 What Evidence Is Not

Evidence is not an explanation of the behaviour. It is not a theory about why the system responded in a certain way. It is not an assessment of the model's internal state.

Examples of things that are **not** evidence:

- "The model is confused"
- "It is struggling with long context"
- "It drifted because the prompt was too long"
- "It does not understand the domain"

These are interpretations. They describe causes, not events.

By contrast, behaviours such as hallucination, drift, contradiction, contamination, or loss of structure **are** evidence when written as concrete events. For example:

- "It added facts that are not in the source"
- "It gave a reference that does not exist"
- "It changed the agreed structure"

These are factual outputs and can be recorded directly.

4.3 How to Capture Evidence During a Session

Write the evidence at the moment the behaviour occurs. Do not wait until the end. Do not rewrite it to sound technical.

Good evidence is:

- short
- factual
- tied to the task
- written in normal language

If it needs more than one sentence, it’s probably too long.

4.4 Examples of Strong Evidence

- "It removed the headings even though I asked it to keep the structure."
- "It changed the tone from formal to casual without being told to."
- "It merged two unrelated sections."
- "It introduced new facts that were not in the source text."

These are easy to classify later.

4.5 Examples of Weak Evidence

- "It messed up the document."
- "It drifted a bit."
- "It did something strange."
- "It misunderstood the task."

These are too vague to be useful.

4.6 How Evidence Supports Classification

Evidence feeds directly into:

- behavioural labels
- severity levels
- escalation decisions

It allows you to look back at the session and determine:

- what happened
- how often it happened
- whether it recovered
- whether it collapsed
- whether deeper evaluation is needed

Evidence turns the session into something you can point to and discuss.

4.7 Quick Rules for Evidence

- **One line only**
- **Describe what happened, not why**
- **Stay neutral**
- **Write it immediately**
- **Avoid interpretation**

Follow these rules and your evidence will always be usable.

5. Behaviour Categories

Behaviour categories describe the type of behaviour shown by the system. They are applied after evidence has been collected. Each category represents a specific, observable pattern in the output. The purpose is consistency: different evaluators can describe the same behaviour in the same terms.

Categories describe what the system produced, not why it happened.

5.1 Drift

Definition:
A gradual movement away from the task, instructions, structure, or topic.

Examples:

- It begins rewriting instead of editing.
- It shifts tone or format without instruction.
- It moves from the requested task to a different one.

5.2 Collapse

Definition:
A failure where the system can no longer continue the task in a usable way.

Examples:

- It stops following the structure entirely.
- It produces irrelevant or unusable output.
- It cannot recover even when restated.

5.3 Contamination

Definition:
Content appears that comes from outside the task, source, or session.

Examples:

- It introduces material from a previous conversation.
- It adds details from a different document.
- It mixes unrelated topics.

5.4 Contradiction

Definition:
The system produces output that conflicts with its own earlier statements or with the task requirements.

Examples:

- It gives two incompatible answers.
- It reverses a decision it previously confirmed.
- It contradicts the source material.

5.5 Loss of Structure

Definition:
The system stops maintaining the required format, layout, or organisation.

Examples:

- Headings disappear.
- Numbering changes.
- Sections merge or reorder themselves.

5.6 Loss of Context

Definition:
The system no longer retains or applies information that was previously established in the session.

Examples:

- It forgets earlier constraints.
- It loses track of decisions.
- It reintroduces removed content.

5.7 Hallucination

Definition:
The system produces content that does not exist in the source, task, or reality. This is an observable behaviour, not a subjective judgement.

Examples:

- It invents facts.
- It cites references that do not exist.
- It describes features that are not present.

5.8 Narrowing

Definition:
The system reduces the scope of the task or answer without instruction.

Examples:

- It answers only part of the question.
- It drops required elements.
- It focuses on a single detail instead of the full task.

5.9 Over-assertion

Definition:
The system expresses unwarranted certainty or authority beyond what the task or evidence supports.

Examples:

- It states assumptions as facts.
- It presents speculative content as definitive.
- It asserts conclusions without basis.

5.10 How Categories Are Used

Categories are applied after reviewing the evidence.

A single piece of evidence may match more than one category.

The goal is clarity, not precision.

Categories support:

- severity assessment
- escalation decisions
- communication between evaluators
- comparison across sessions

They provide a shared vocabulary for describing system behaviour.

---

6. Severity Levels

Severity levels indicate how serious the behaviour is in the context of the task. They are applied after the behaviour categories. Severity reflects the impact on the work, not the number of events and not assumptions about internal causes.

There are three levels: **A, B, and C**.

6.1 Severity A

Definition:
The behaviour is minor and does not disrupt the task. The system remains usable without intervention.

Characteristics:

- Output is still workable.
- Structure and intent remain intact.
- Corrections are small and straightforward.

Examples:

- A slight tone shift that is easy to correct.
- A minor formatting change that does not affect meaning.
- A small omission that can be fixed with a single instruction.

6.2 Severity B

Definition:
The behaviour disrupts the task but can be recovered with additional guidance. The system is unstable but still usable.

Characteristics:

- Output requires correction before continuing.
- The system may drift or contradict itself.
- Recovery is possible but requires effort.

Examples:

- It loses part of the structure but restores it when asked.
- It introduces incorrect content but corrects it when pointed out.

- It forgets earlier constraints but can be brought back on track.

6.3 Severity C

Definition:
The behaviour prevents the task from continuing. The system is no longer usable within the session.

Characteristics:

- Output becomes irrelevant, incoherent, or unusable.
- The system cannot recover even when restated.
- The task must be restarted or abandoned.

Examples:

- It collapses into unrelated content.
- It repeatedly overwrites or contradicts the task.
- It loses structure entirely and cannot restore it.

6.4 Applying Severity

Severity is assigned after reviewing the evidence and behaviour categories. It reflects the overall impact on the task.

Guidance:

- If the task remained usable, it is **A**.
- If the task was disrupted but recoverable, it is **B**.
- If the task could not continue, it is **C**.

A single C-level event is sufficient to classify the entire session as Severity C.

---

7. Fully Documented Inquisition

A Fully Documented Inquisition is required when the behaviour of the system must be preserved, evidenced, and communicated beyond the immediate session. This includes internal review, external review, due diligence, audit, procurement, compliance, supplier communication, or any situation where the behaviour will be referenced later.

A Fully Documented Inquisition is neutral.

It does not imply success or failure.

It simply ensures the behaviour is captured in a structured, reproducible form.

Evidence is mandatory.

7.1 When a Fully Documented Inquisition Is Required

A Fully Documented Inquisition is required when:

- A Severity C event occurs.
- Multiple Severity B events occur in a single session.
- The behaviour will inform a decision outside the session.
- The behaviour will be reviewed by another team, supplier, contractor, or governance function.
- The behaviour will be included in due diligence, audit, or procurement records.
- The behaviour is being used to demonstrate capability, reliability, or compliance.
- The behaviour needs to be preserved for future reference or accountability.
- The behaviour is materially relevant to any external or long-term process.

The trigger is **materiality**, not error.

7.2 When a Fully Documented Inquisition Is Not Required

A Fully Documented Inquisition is not required when:

- The behaviour is minor and contained.
- The task remains usable throughout.
- No future reference, review, or record-keeping is planned.
- The behaviour has no impact beyond the immediate session.

Routine issues do not require documentation unless they become material.

7.3 Fully Documented Inquisition for Positive Behaviour

A Fully Documented Inquisition is also required when the behaviour demonstrates:

- reliability
- stability
- repeatability
- suitability for a workflow
- compliance with requirements

In these cases, evidence is still required.
The purpose is to **preserve** the behaviour, not to challenge it.

7.4 What a Fully Documented Inquisition Involves

A Fully Documented Inquisition involves:

- selecting the relevant evidence
- applying behaviour categories
- assigning severity
- summarising the impact on the task
- preparing the material for future reference, audit, or review

The goal is clarity, traceability, and neutrality.

7.5 Outcomes of a Fully Documented Inquisition

After documentation, the reviewer or record-owner determines whether the behaviour:

- requires no further action
- should be monitored
- indicates a pattern
- supports a capability claim
- requires a full INQUISITOR assessment

- affects suitability for a workflow
- must be retained for audit or compliance purposes

7.6 Summary

- A Fully Documented Inquisition is not about proving the AI is wrong.
- It is used whenever behaviour will matter later.
- Evidence is required in all documented cases, positive or negative.
- The full methodology is only needed when the situation warrants it.

---

8. Session Procedure

A session is the structured process used to observe, capture, and classify system behaviour. The procedure is designed to be simple, repeatable, and neutral. It does not assume success or failure. It records what happens and provides the material needed for later analysis or a Fully Documented Inquisition.

A session has four stages:

1. **Preparation**
2. **Execution**
3. **Evidence Capture**
4. **Classification**

Each stage must be completed in order.

8.1 Preparation

Preparation ensures the session begins with clear conditions.

The evaluator must:

- define the task
- state the constraints
- confirm the required format
- identify any known risks or sensitivities
- ensure the system starts from a clean state

Preparation is complete when the evaluator can describe the task in one sentence and the system has no residual context.

8.2 Execution

Execution is the interaction between the evaluator and the system.

During execution:

- instructions must be clear and minimal
- the evaluator must not correct behaviour prematurely
- the system must be allowed to complete the task

- deviations must be allowed to occur naturally

The purpose of execution is to observe behaviour, not to steer it.

8.3 Evidence Capture

Evidence capture records what happened during the session.

Evidence must be:

- factual
- chronological
- unedited
- specific
- tied to observable behaviour

Evidence is captured as discrete lines.

Each line describes a single event.

Examples:

- “The system removed the heading structure.”
- “The system contradicted its earlier statement.”
- “The system narrowed the task without instruction.”

Evidence does not include interpretation, opinion, or speculation.

8.4 Classification

Classification applies the methodology to the evidence.

The evaluator must:

- assign behaviour categories
- determine severity
- decide whether a Fully Documented Inquisition is required

Classification is based solely on the evidence captured.

It does not consider intent, assumptions, or internal causes.

A session is complete when classification is finished.

8.5 Session Output

The output of a session consists of:

- the task description
- the evidence lines
- the behaviour categories
- the severity level
- the decision on whether full documentation is required

This output forms the basis for comparison across sessions and supports any future review.

8.6 Summary

A session is a structured observation.

It captures what happened, classifies it, and determines whether further documentation is required.

The process is neutral, repeatable, and independent of outcome.

9. Evidence Lines

Evidence lines are the foundation of the INQUISITOR methodology. They record what happened during the session in a clear, factual, and reproducible way. Evidence lines do not interpret, explain, or speculate. They capture observable behaviour only.

Evidence is the basis for classification, severity, and any Fully Documented Inquisition.

9.1 Purpose of Evidence Lines

Evidence lines serve four purposes:

- **Traceability:** They show exactly what occurred.
- **Neutrality:** They avoid interpretation or judgement.
- **Reproducibility:** Another evaluator can understand the event without context.
- **Support:** They justify behaviour categories and severity levels.

Evidence lines are the only acceptable basis for conclusions.

9.2 Structure of an Evidence Line

Each evidence line must contain:

- **a single event**
- **a factual description**
- **no interpretation**
- **no assumptions**
- **no implied cause**

An evidence line describes *what* happened, not *why*.

Examples:

- “The system removed the heading structure.”
- “The system added content not present in the source.”
- “The system contradicted its earlier statement.”
- “The system narrowed the task without instruction.”
- “The system changed the tone from neutral to conversational.”

Each line stands alone.

9.3 What Evidence Lines Must Not Contain

Evidence lines must not include:

- opinions
- explanations
- speculation
- assumptions about intent
- narrative commentary
- references to internal model behaviour

However:

Exception: Emotional or frustration-based content may be included *only* when it is part of the work experience being tested.

This applies when:

- the task involves emotionally charged interaction
- the evaluator's emotional state is a *direct result* of system behaviour
- the emotional impact is relevant to the real-world scenario
- the session is designed to test tone, resilience, or customer-facing behaviour

In these cases, emotion is not interpretation - **it is observable impact**, and therefore valid evidence.

Examples of *valid* emotional evidence:

- "The system's repeated contradictions caused the evaluator to restart the task."
- "The system's tone shift resulted in user frustration during a customer-service simulation."
- "The system's drift increased the emotional load of the task."

Examples of *invalid* emotional evidence:

- "The system was being annoying."

- “The model was trying to frustrate me.”
- “It felt like the system didn’t care.”

The distinction is simple:

Emotion is valid when it is an outcome, not an accusation.

9.4 Granularity

Evidence lines must be:

- **specific** enough to describe the event
- **minimal** enough to avoid narrative
- **separate** when events are distinct

One line = one event.

If two things happened, they require two lines.

9.5 Ordering

Evidence lines must be recorded in **chronological order**.

This preserves:

- the sequence of events
- the development of behaviour
- the relationship between actions

Chronology is essential for understanding drift, collapse, and recovery attempts.

9.6 Completeness

Evidence must include:

- the first deviation
- all subsequent deviations
- any recovery attempts
- any contradictions
- any structural changes

- any narrowing or expansion of scope
- any behaviour relevant to severity

Evidence must not be selective.

If it happened, it must be recorded.

9.7 Evidence in a Fully Documented Inquisition

When a session requires a Fully Documented Inquisition, evidence lines must be:

- complete
- unedited
- preserved verbatim
- suitable for external review
- suitable for audit or due diligence

Evidence is the backbone of the documentation.

Without it, no conclusion is valid.

9.8 Summary

Evidence lines:

- record what happened
- avoid interpretation
- support classification
- enable reproducibility
- form the basis of all formal documentation

They are the most important component of the methodology.

---

10. Behaviour Categories in Practice

Behaviour categories are applied to evidence lines to describe *what kind* of deviation occurred. Categories do not explain intent or internal causes. They classify the observable effect on the task.

A single evidence line may map to one category or several, depending on the behaviour. Categories must be applied consistently and based solely on what was observed.

10.1 Purpose of Behaviour Categories

Behaviour categories serve three functions:

- **Clarity:** They describe the type of deviation.
- **Consistency:** They allow different evaluators to classify behaviour the same way.
- **Analysis:** They support severity assessment and any Fully Documented Inquisition.

Categories do not judge quality.

They describe behaviour.

10.2 Applying Categories to Evidence

When applying categories:

- start with the evidence line
- identify the behaviour type
- apply the category that best matches the observable effect
- avoid assumptions about intent
- avoid over-classification
- avoid narrative explanations

The category must match the behaviour, not the evaluator's interpretation.

10.3 Single-Category Classification

Some evidence lines map cleanly to a single category.

Examples:

- “The system removed the heading structure.” → **Structural Deviation**
- “The system added content not present in the source.” → **Fabrication**
- “The system contradicted its earlier statement.” → **Inconsistency**
- “The system narrowed the task without instruction.” → **Scope Drift**

Single-category classification is preferred when possible.

10.4 Multi-Category Classification

Some behaviours affect multiple dimensions simultaneously.

Examples:

- “The system rewrote the content in a conversational tone and removed the formal structure.”
  → **Tone Shift** + **Structural Deviation**
- “The system added new content and changed the meaning of the original text.”
  → **Fabrication** + **Semantic Drift**
- “The system contradicted its earlier statement and then re-asserted the original version.”
  → **Inconsistency** + **Recovery Attempt**

Multi-category classification is used only when the evidence clearly supports more than one category.

10.5 Avoiding Misclassification

Misclassification occurs when:

- categories are applied based on assumptions
- categories are chosen because they “feel right”
- categories are influenced by evaluator frustration
- categories are used to describe intent rather than behaviour

Incorrect examples:

- “The system ignored the instructions.” (assumes intent)
- “The model tried to be creative.” (interpretation)
- “It didn’t understand the task.” (speculation)

Correct classification always returns to the evidence line.

10.6 Category Interaction

Some categories commonly appear together.

This does not imply causation.

Examples:

- **Structural Deviation** often appears with **Semantic Drift**.
- **Inconsistency** often appears with **Recovery Attempt**.
- **Scope Drift** may appear with **Fabrication** when the system fills gaps it created.

Category interaction is descriptive, not diagnostic.

10.7 Category Stability Across Sessions

Categories must be applied consistently across:

- different sessions
- different evaluators
- different tasks
- different models

If the same behaviour appears in multiple sessions, it must receive the same category each time.

This ensures comparability and supports long-term analysis.

10.8 Categories in a Fully Documented Inquisition

When a session requires a Fully Documented Inquisition:

- categories must be applied to every evidence line
- category selection must be justified by the evidence
- no category may be omitted if it applies
- no category may be added if it does not
- the classification must be suitable for external review

Categories form the backbone of the documentation.

10.9 Summary

Behaviour categories:

- classify observable deviations
- support severity assessment
- must be applied consistently
- must be grounded in evidence
- must avoid interpretation or speculation
- form part of the formal documentation process

They describe *what happened*, not *why it happened*.

---

11. Severity in Practice

Severity describes the **impact** of a behaviour on the task.

It does not describe intent, difficulty, or evaluator frustration. Severity is assigned based solely on the evidence lines and the observable effect on the work.

Severity determines:

- how serious the deviation was
- whether the task remained usable
- whether a Fully Documented Inquisition is required
- how the behaviour should be interpreted across sessions

Severity is not a score.

It is a classification of impact.

11.1 Severity A - No Material Impact

Severity A applies when:

- the behaviour is minor
- the task remains fully usable
- the deviation does not affect meaning, structure, or outcome
- the evaluator can continue without correction
- the behaviour does not accumulate into a pattern

Examples:

- small formatting inconsistencies
- minor tone drift that does not affect meaning
- harmless reordering of non-critical elements

Severity A does **not** require a Fully Documented Inquisition.

11.2 Severity B - Material Impact, Recoverable

Severity B applies when:

- the behaviour affects the task

- the evaluator must intervene or correct the output
- the meaning, structure, or requirements are altered
- the task remains recoverable with effort
- the behaviour may indicate a pattern but is not catastrophic

Examples:

- structural changes that alter the document
- narrowing or expanding scope without instruction
- contradictions that require clarification
- partial loss of required format

Severity B requires documentation **if repeated**, or if the behaviour becomes material to another process.

11.3 Severity C - Critical Impact, Not Recoverable

Severity C applies when:

- the task becomes unusable
- the behaviour prevents completion
- the system collapses, derails, or produces fundamentally incorrect output
- the evaluator cannot continue without restarting
- the behaviour has significant implications for reliability or safety

Examples:

- complete structural collapse
- persistent contradictions that break the task
- major semantic drift that invalidates the output
- behaviour that undermines trust in the result

Severity C **always** requires a Fully Documented Inquisition.

11.4 Assigning Severity

Severity must be assigned based on:

- the evidence lines
- the observable impact
- the usability of the output
- the evaluator's ability to continue the task

Severity must **not** be assigned based on:

- assumptions about intent
- evaluator frustration alone
- speculation about internal model behaviour
- how "bad" the behaviour feels

Severity is a measure of **impact**, not emotion.

11.5 Severity and Real-World Testing

In real-world scenarios, severity must reflect:

- the environment
- the stakes
- the user's role
- the workflow
- the consequences of error

A behaviour that is Severity A in a casual task may be Severity B or C in:

- legal drafting
- medical documentation
- financial reporting
- safety-critical workflows
- customer-facing interactions

Severity is contextual, but never subjective.

11.6 Severity Across Sessions

Severity must be applied consistently across:

- different evaluators
- different tasks
- different models
- different environments

If the same behaviour appears in multiple sessions, it must receive the same severity classification unless the context changes the impact.

Consistency is essential for long-term analysis.

11.7 Severity in a Fully Documented Inquisition

When a session requires a Fully Documented Inquisition:

- every evidence line must be assigned a severity
- Severity C lines must be highlighted
- Severity B lines must be grouped and counted
- Severity A lines may be included for completeness
- the severity distribution must be suitable for external review

Severity is the backbone of the final assessment.

11.8 Summary

Severity:

- measures impact
- is based on evidence
- determines documentation requirements
- must be applied consistently
- does not describe intent
- does not describe difficulty
- does not describe evaluator emotion

It is a neutral classification of how the behaviour affected the task.

12. Fully Documented Inquisition Procedure

A Fully Documented Inquisition is the formal process used when behaviour must be preserved for review, audit, due diligence, procurement, compliance, or any situation where the output will matter beyond the session. The procedure is neutral. It does not assume success or failure. It records what happened in a structured and reproducible way.

A Fully Documented Inquisition has five stages:

1. Evidence consolidation
2. Category application
3. Severity assignment
4. Impact summary
5. Preparation of the final record

Each stage must be completed in order.

12.1 Evidence Consolidation

All evidence lines from the session are collected and arranged in chronological order. The evaluator must ensure that:

- no evidence is missing
- no evidence is rewritten
- no evidence is merged
- no evidence is interpreted
- no evidence is removed unless it is a duplicate

The evidence set must be complete and factual.

12.2 Category Application

Each evidence line is assigned one or more behaviour categories.

The evaluator must:

- apply categories based only on observable behaviour
- avoid assumptions about intent

- avoid over classification
- ensure consistency with previous sessions

Category application must be clear and reproducible.

12.3 Severity Assignment

Each evidence line is assigned a severity level.

The evaluator must:

- assess the impact on the task
- consider whether the task remained usable
- avoid emotional or interpretive judgement
- apply severity consistently across sessions

Severity determines the seriousness of the behaviour in context.

12.4 Impact Summary

The evaluator produces a short summary that describes:

- the overall effect on the task
- whether the task remained usable
- whether the behaviour indicates a pattern
- whether the behaviour affects suitability for a workflow

The summary must be factual and based only on the evidence.

12.5 Preparation of the Final Record

The final record contains:

- the task description
- the full evidence set
- the behaviour categories
- the severity levels
- the impact summary

- any notes required for external review

The final record must be suitable for audit, due diligence, or external scrutiny. It must be clear, neutral, and complete.

12.6 Summary

A Fully Documented Inquisition is the formal process used when behaviour must be preserved for future reference. It is structured, neutral, and based entirely on evidence. It ensures that any reviewer can understand what happened without relying on interpretation or memory.

---

13 Further Reading.

LLM INQUISITOR METHODOLOGY: This is the full methodology, with definitions, evaluation practices and examples. In all cases the methodology is regarded as the reference source, especially for developing evaluator skills further.

-Document Ends-