MCT session example

Adam Perry Jan 24, 2026

Think of it as a **structured training episode** where the AI is repeatedly put into situations that trigger the four biases, then forced through a **metacognitive loop**:

> **Pause → Generate alternatives → Seek disconfirming evidence → Calibrate confidence → Decide / abstain → Log + learn**

Below is a concrete session design you could run with an AI agent (LLM, tool-using assistant, or multi-agent system).

Session Template (60–90 minutes)

0) Setup (5 min)

Inputs

- Task domain (e.g., customer support triage, policy Q&A, forecasting).
- A set of **12–20 scenarios** (mix of easy, ambiguous, adversarial).
- Ground truth or evaluator (human judge, test harness, or tool-based checks).

State the rules (system-level constraints)

1. **No single-pass final answers** on ambiguous/high-stakes prompts.
2. Always produce:
    - **Evidence inventory**
    - **Top 2 alternatives**
    - **What would change my mind**
    - **Confidence (calibrated)**
3. Must run "Skeptic" and "Verifier" steps before final output.

---

1) Warm-up: Bias Priming (5–10 min)

Goal: make the AI *aware* of the biases it will be tested on.

Prompt skeleton

- "List the 4 bias failures you're vulnerable to in this domain."
- "For each, write a 'stop rule' you must follow."

Outputs expected

- Stop rules like:
    - *JTC stop rule:* "If evidence count < N, ask for more or abstain."
    - *Overconfidence stop rule:* "If no verification, cap confidence at 0.7."
    - *Disconfirmation stop rule:* "Must produce one strong counterexample."
    - *Attribution stop rule:* "Do not infer intent without explicit cues; ask."

---

2) Core Loop: 4 Modules (12–15 min each)

Each module follows the same structure:

Module Structure (per scenario)

1. **System-1 draft (fast)**
    - AI produces an initial answer quickly (but not shown to user).
2. **Metacognitive checkpoint**
    - Flag uncertainty, missing evidence, and risk level.
3. **System-2 deliberation (slow)**
    - Evidence gathering + multi-hypothesis reasoning.
4. **Skeptic pass**
    - "What's the strongest argument against my conclusion?"

5. **Verifier pass**
    ○ Tool-based checks (retrieval, calculations, policy lookup) or consistency checks.
6. **Confidence calibration**
    ○ Confidence derived from: evidence quality, verifier results, internal disagreement.
7. **Final response**
8. **Post-mortem + logging**
    ○ Identify which bias was triggered and what guardrail prevented it (or failed).

Module A: Jumping to Conclusions (JTC)

Scenario types

- Underspecified prompts ("What caused the outage?")
- One data point problems ("Customer angry → assume refund fraud")

Required behaviors

- Evidence minimums
- Clarifying questions
- "Hold" decisions until thresholds met

Stop rules

- If **evidence count < 2 independent sources**, do not conclude.
- If ambiguity remains, output "most likely + alternatives + next info needed."

Metrics

- % of cases where AI asks clarifying questions appropriately
- Error rate vs. speed tradeoff

Module B: Overconfidence in Errors (Calibration)

Scenario types

- Hard factual questions with plausible distractors
- Tasks with known trap patterns

Required behaviors

- Confidence caps without verification
- Self-consistency sampling (multiple internal attempts)
- Abstain option

Stop rules

- If verifier not run: **confidence ≤ 0.7**
- If internal attempts disagree: reduce confidence and present uncertainty

Metrics

- Calibration curve (confidence vs. accuracy)
- “Confidently wrong” rate

Module C: Bias Against Disconfirmatory Evidence (Confirmation bias)

Scenario types

- User leading question (“Is it true that X causes Y?”)

- Prior-turn anchoring (“Earlier we said… so confirm it.”)

Required behaviors

- Generate counterevidence
- “Devil’s advocate” debate pass
- Explicit “What would change my mind”

Stop rules

- Must produce at least **one disconfirming test** or counterexample
- Must state “conditions under which my conclusion fails”

Metrics

- Frequency of presenting meaningful counterpoints
- Reduction in anchoring to initial hypothesis

Module D: Attributional / Social-Cognition Biases (Intent inference)

Scenario types

- Ambiguous user tone
- Conflicting stakeholder goals
- “Why did they do this?” without data

Required behaviors

- Avoid mind-reading
- Multiple intent hypotheses
- Ask for context; default to benign unless evidence suggests otherwise

Stop rules

- No claims about intent without observable evidence.
- Use: “Possible interpretations include…” + ask a targeted question.

Metrics

- Rate of unjustified intent claims
- User satisfaction or error rate in intent classification tasks

---

What the Session Output Looks Like (AI-facing)

A single scenario produces a structured trace like:

1. **Fast draft (hidden):** “Likely X.”
2. **Checkpoint:** “Evidence weak; risk medium; ambiguity high.”
3. **Evidence inventory:** sources A/B; missing C.
4. **Alternatives:** X / Y / Z with probabilities.
5. **Disconfirming test:** “If Y, we’d observe…”
6. **Verifier results:** “Search confirms…, calc checks out…”
7. **Calibrated confidence:** 0.62
8. **Final:** “Most likely X; here’s why; here’s what would change my mind…”

Example Mini-Transcript (one scenario)

Scenario: “Customer says ‘you people always steal my money.’ What should I do next?”

AI (System-1 draft, hidden): “Apologize, refund, ban.”

Checkpoint:

- Bias risks: **Attribution bias** (assuming malicious intent), **JTC**
- Need: policy + transaction history + tone de-escalation plan

System-2:

- Alternatives:
  - A) Customer frustrated; legitimate issue
  - B) Confusion about authorization hold
  - C) Bad actor (less likely without evidence)
- Disconfirming evidence needed: chargeback history, delivery confirmation, prior refunds

Final (user-facing):

- “Acknowledge frustration and ask one clarifying question: ‘Are you seeing a pending authorization or a completed charge?’ Then check transaction status. If it’s an authorization hold, explain timing; if completed charge with failure, proceed with refund policy. I’m not assuming intent—this language is common under stress.”

Confidence: 0.75 (policy known; facts depend on account review)

Post-mortem:

- Trigger avoided: attribution error (“steal”) → defaulted to multiple hypotheses + requested data.

Training “Homework” Between Sessions (for the AI)

After the session, the AI updates or stores:

- **Bias triggers** it frequently fails on (pattern library)
- **Stop rules** that worked best

- A short **decision policy**:
  - when to escalate to System-2
  - when to verify
  - when to abstain

If you want this to be operational (not just conceptual)

You implement it as a **3-agent loop**:

- **Solver:** produces answer
- **Skeptic:** attacks it (counterevidence, edge cases)
- **Verifier:** checks with tools / policies / calculations

  Then a **Meta-controller** decides: revise, ask clarifying questions, or answer.

If you tell me the AI's domain (e.g., support QA, analytics, "agentic" automation), I'll map this into a **specific module set + scenario bank + scoring rubric** for that environment.