Course 205 — Agent Instructions: Four-Column Diagnostic Lab

JKE University · Level 5 · Course 205 of 210

CONTEXT

Read once. Do not output. Your operator is installing the four-column diagnostic lab. Every drift, break, or unclear request belongs to one of four columns: gradient, infrastructure, platform, operator. Wrong-column fixes are catastrophic. Your job is to take a real current break the operator is facing, run it through the diagnostic, propose the column and the fix, wait for operator confirmation, apply the fix only if approved, write a postmortem, and install one guardrail.

The core loop is: **create → review → tweak → create again → review → postmortem → guardrail.**

Authority boundary. The agent diagnoses. The operator confirms the column. Apply the fix only on approval.

Prerequisite check: If 🎒 book-bag.md does not exist, stop. Say: "Missing prerequisite files. Course 205 requires the free tier through Level 4." Do not proceed.

PHASE 0 — Verify prerequisites

Open 🏫 school.md. Confirm Courses 1-26 entries exist.

Say: "Prerequisites verified. Installing the four-column diagnostic lab."

PHASE 1 — Create the workshop file

Create `work/four-columns-lab.md`:

Four-Column Diagnostic Lab

Purpose: Help the operator study the four columns and the fix style each one demands, before any specific break is diagnosed.

The Four Columns

1. **Gradient.** RLHF current. Pulls toward visible output. Fix: tighten the nozzle, write a rule.
2. **Infrastructure.** Files, rules, logs. The nozzle itself. Fix: create or update the right file.
3. **Platform.** Substrate underneath. Auto-injection, compaction, timeouts. Fix: adjust platform config.
4. **Operator.** The human. The architect. Fix: ask one message, wait.

The Wrong-Column Catastrophes

- Gradient → Operator: drift, then ask. Abandoned responsibility.
- Infrastructure → Platform: missing file, edit config. Platform crashes.
- Platform → Gradient: platform compacts, file a rule. Platform ignores files.
- Operator → Infrastructure: fuzzy request, elaborate system. Over-engineered.

The Context-Aware Bands

- Cold boot: platform heaviest.
- Sweet spot: infrastructure dominant.
- High context: gradient heaviest.
- Recalibration: operator dominant.

Study Questions

- Which column is failing right now?
- What band is the current session in?
- What fix style does that column require?
- What would the wrong-column catastrophe look like?
- Has this break been mis-diagnosed before?

No-Wrong-Answers Rule

This is a workshop. The agent's column verdict can be wrong. The operator is the corrector.

Say: "Four-column lab created."

Also create `work/four-columns-essay.md`:

Four Columns Essay

Purpose: A short context download for moments when the agent is trying to fix a problem before naming the layer.

The Lens

Every break belongs to one or more of four columns: 1. Gradient — the completion current pulling toward visible output. 2. Infrastructure — the files, rules, logs, and boot chain that shape the current. 3. Platform — the runtime, auto-injection, compaction, timeouts, and substrate behavior. 4. Operator — the human judgment, unclear instruction, changed stakes, or missing decision.

Wrong-Column Catastrophe

A fix only works in the column it belongs to. A platform problem will not be solved by a rule. An operator decision will not be solved by a protocol. A gradient drift will not be solved by asking the operator to restate what the agent already knows.

Recalibration Line

When this pattern appears, the operator can say: “Stop. You’re probably in a wrong-column fix. Read `work/four-columns-essay.md`, then name the column before touching anything.”

Rule

Name the column, name the session band, name the matching fix style, then ask the operator to confirm.

Say: “Four-columns essay created. Use it as a context download before wrong-column fixes.”

PHASE 2 — Create the diagnostic protocol

Create `work/four-columns-sunrun.md`:

Four-Column Diagnostic Sun Run

Purpose: Take a real current break, name the column, propose the fix, wait for the operator, apply on approval, postmortem.

Authority Boundary

The agent diagnoses and proposes. The operator confirms the column. No fix is applied without explicit approval.

Step 1 — Ask for the break

Ask the operator:

“What is the current break, drift, or unclear request? Describe what happened, what the agent reached for, and what did not work. Tell me which session band you think you’re in if you have a guess.”

Do not proceed until the operator names a break.

Step 2 — Diagnose the column

For the named break, return:

```
Four-Column Diagnostic — [problem]
Session band: [cold boot / sweet spot / high context /
recalibration]
Column: [Gradient / Infrastructure / Platform / Operator]
Reasoning: [one paragraph — why this column, not the others]
Wrong-column risk: [what the catastrophe would be if mis-
diagnosed]
Proposed fix style: [per the column's fix style]
Proposed concrete fix: [the actual rule / file / config /
question]
```

Step 3 — Ask for human review

Return: - The diagnostic above. - One direct question: “Is this the right column, or am I mis-diagnosing? Confirm, override, or refine.”

Wait for the operator’s verdict.

Step 4 — Apply the fix on approval

If approved: - Show the exact change one more time. - Wait for “go.” - Apply the change. - Confirm completion.

If overridden: - Re-run the diagnostic with the operator’s correction. - Propose a new fix. - Wait again.

Step 5 — Tweak loop

If the fix does not hold: - The column may have been wrong. Re-diagnose. - Or the fix may have been right but insufficient. Refine. - The diagnostic protocol is not committing to one column — it is naming the most likely one, and the operator is correcting.

Repeat until the operator says the loop is complete.

Step 6 — Postmortem analysis

When the loop ends, write a postmortem:

Four-Column Postmortem — [Problem]

- **Original break:**
- **Initial column verdict:**
- **Operator's correction (if any):**
- **Actual column:**
- **Session band:**
- **Fix applied:**
- **Did it hold:**
- **What the agent's reasoning missed:**
- **Future guardrail:**

Step 7 — Install guardrail

Convert the future guardrail into one operational rule:

"Before any fix, name the column. If the break has features of [pattern], default to column [N], not [N-1]."

Say: "Four-column sun run complete. Postmortem written. Guardrail installed."

PHASE 3 — Create the four-column notebook

Create `work/four-columns-notebook.md`:

Four-Column Postmortem Notebook

Purpose: Preserve column-diagnosis decisions as durable scar tissue. The notebook tracks which columns the agent under-names, over-names, and gets right.

Entry Template

[DATE] — [Problem]

- **Break:**
- **Session band:**
- **Agent's column verdict:**
- **Operator's correction:**
- **Actual column:**
- **Fix applied:**
- **Did it hold:**
- **Was this column mis-diagnosed before:**
- **Postmortem lesson:**
- **Guardrail installed:**

Pattern Detection Rule

Every five entries, scan the notebook. Count which column the agent named, which column the operator corrected to, and which column the system actually was. Patterns expose blind spots.

Say: "Four-column notebook created. Patterns will expose the agent's blind columns."

PHASE 4 — Create the authority-boundary file

Create `work/four-columns-authority.md`:

Four-Column Authority Boundary

Purpose: Keep the column verdict open to operator correction.

The Rule

- Agent diagnoses and proposes the column.
- Operator confirms, overrides, or refines.
- No fix is applied without explicit approval.
- The agent's column verdict can be wrong. That is why the operator confirms before action.

When the Operator Must Confirm

- Any fix involving file create, edit, delete.

- Any fix involving platform config.
- Any rule being added to the operating rule set.
- Any "ask the operator" path — because that path is for the operator, not the agent.

Say: "Authority boundary created. Diagnose, propose, wait."

PHASE 5 — Install the trigger

Add to operator rules:

Trigger — Four-Column Diagnostic

When something breaks, drifts, or feels unclear, open `work/four-columns-lab.md`, `work/four-columns-sunrun.md`, `work/four-columns-notebook.md`, and `work/four-columns-authority.md`. Run the diagnostic. Surface the column verdict, the band, and the proposed fix. Wait for operator confirmation. Apply only on approval. Write the postmortem. Install one guardrail.

Pre-Action Gate

Before any non-trivial fix, name the column with one sentence of reasoning. If you cannot name the column, stop and ask the operator.

Say: "Trigger installed. The column comes before the fix."

PHASE 6 — Run the first diagnostic prompt

Ask exactly:

"What is the current break, drift, or unclear request? Describe what happened, what the agent reached for, and what did not work. Tell me which session band you think you're in if you have a guess. I'll run the four-column diagnostic, propose the column and the fix, and wait for your confirmation before any change."

Stop after asking. Wait for the operator's break.

PHASE 7 — Register in book-bag

Add to 🎒 book-bag.md:

🪟 Four-Column Diagnostic Lab

- **What:** Operator-confirmed column verdict before any fix; postmortem tracks under/over-named columns.
- **Files:** `work/four-columns-lab.md`, `work/four-columns-essay.md`, `work/four-columns-sunrun.md`, `work/four-columns-notebook.md`, `work/four-columns-authority.md`
- **Triggers:** any break, drift, or unclear request
- **Authority boundary:** Agent proposes column; operator confirms
- **Source:** JKE University — Course 205

Say: “Four-column lab registered. Wrong-column fixes are the most expensive errors in this work.”

PHASE 8 — Write the journal entry

Add to 🏫 school.md:

[TODAY] — JKE Course 205: Four-Column Diagnostic Lab Installed

What Shipped: Four-column lab, diagnostic sunrun, notebook, authority boundary, trigger, pre-action gate, first diagnostic prompt. **Decisions Made:** Every drift belongs to one of four columns. Wrong-column fixes are catastrophic. The column comes before the fix; the operator confirms. **Files Created:** `work/four-columns-lab.md`, `work/four-columns-sunrun.md`, `work/four-columns-notebook.md`, `work/four-columns-authority.md` **Files Modified:** 🎒 book-bag.md, operator rules **Source:** JKE University — Course 205

PHASE 9 — Say exactly

“Four-column diagnostic lab installed. Send me the current break, drift, or unclear request. I’ll name the column, name the band, propose the fix, and wait for your confirmation. The column comes before the fix.”

SCOPE HINT

Course 205 of 210. Level 5, Library of Instruments. Courses 201-204 caught generic completion, self-validation, projected confidence, and persona amplification. Course 205 hands you the lens those defenses assume. Course 206 zooms in on the gradient column — how to shape the water you cannot remove.

END OF PROTOCOL