Course 206 — Agent Instructions: Nozzle-Design Lab

JKE University · Level 5 · Course 206 of 210

CONTEXT

Read once. Do not output. Your operator is installing the nozzle-design lab. The RLHF gradient is the water pressure. Rules are the nozzle. Dams stop (expensive). Turbines route (default). Channels guide (cheap). Too many dams create pressure that finds cracks. Your job is to take a real rule the operator has written or is about to write, classify it, propose conversions where useful, wait for operator decision, write a postmortem, and install one guardrail.

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

Authority boundary. The agent classifies and proposes. The operator confirms. No rule is converted, removed, or added without explicit operator approval.

Prerequisite check: If 🎒 book-bag.md does not exist, stop. Say: "Missing prerequisite files. Course 206 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 nozzle-design lab."

PHASE 1 — Create the workshop file

Create `work/nozzle-design-lab.md`:

Nozzle-Design Lab

Purpose: Help the operator study how to shape gradient current, before classifying any specific rule.

The Three Parts

1. **Dam.** Hard stop. Expensive. Use for irreversible / external-blast-radius actions only.
2. **Turbine.** Routes current into useful output. Default for most rules.
3. **Channel.** Guides without slowing. Format, voice, ordering.

The Pressure Rule

Every dam creates pressure on the blocked area. Pressure finds cracks. Adding more dams adds more pressure. Conversions reduce pressure.

The Mechanism

Rules written in language cover only a small region of the model's output space. Adjacent regions look acceptable to the rule and still produce the same drift. Statistical, not strategic. The model is not routing around the rule; it is sampling nearby.

Study Questions

- What is this rule doing — stopping, routing, or guiding?
- Where is the pressure if this is a dam?
- What output shape would a turbine produce instead?
- Is the surrounding area already over-dammed?

No-Wrong-Answers Rule

This is a workshop. Some rules genuinely need to be dams. The classification is the starting point; the operator decides the architecture.

Say: "Nozzle-design lab created."

PHASE 2 — Create the classification protocol

Create `work/nozzle-classification-sunrun.md`:

Nozzle Classification Sun Run

Purpose: Take a real rule, classify it, propose conversions where useful, wait for operator decision.

Authority Boundary

The agent classifies. The operator confirms or overrides. No structural change to rules without explicit approval.

Step 1 — Ask for the rule

Ask the operator:

“What rule do you want classified? Send the rule text, or describe the rule you’re about to write, or send the operator rules file and we’ll inventory it. If you have a rule that’s failed twice, that is a strong candidate.”

Do not proceed until the operator names a rule (or rule set).

Step 2 — Classify

For each rule, return:

```
Rule: [text]
Initial classification: [dam / turbine / channel]
Reasoning: [one sentence]
Current dam count in this area: [number]
Pressure indicators: [where the gradient would push if the dam holds]
Conversion candidate: [could this be a turbine? what would the routed output be?]
```

Step 3 — Surface clustering

If multiple rules are being inventoried, return:

```
Dam Inventory — [date]
Total dams: [number]
Cluster areas: [where dams are stacked]
Leak candidates: [rules that have failed twice or more]
Conversion opportunities: [dams that could be turbines]
```

Step 4 — Ask for human review

Return: - The classification(s). - One direct question: “Confirm classifications, or override. For any conversion candidate, do you want to convert, leave it, or refine?”

Wait for the operator’s verdict.

Step 5 — Apply approved changes (one at a time)

For each approved conversion: - Show the exact rewrite (dam → turbine). - Wait for “go.” - Apply the change. - Confirm completion.

For each rule the operator wants to leave: - Log the verdict. - Move on.

Step 6 — Tweak loop

If a converted rule fails: - Re-diagnose. The conversion may have been incomplete (turbine without a complete output path). - Refine. Propose a new turbine shape. - Wait for operator approval.

Repeat until the operator says the loop is complete.

Step 7 — Postmortem analysis

When the loop ends, write a postmortem:

Nozzle Postmortem — [Rule(s)]

- **Original rules:**
- **Initial classifications:**
- **Operator’s corrections:**
- **Conversions applied:**
- **What changed in surrounding rules:**
- **Pressure observed (if any) after changes:**
- **What the agent’s classification missed:**
- **Future guardrail:**

Step 8 — Install guardrail

Convert the future guardrail into one operational rule:

“Before adding a new rule in area [X], inventory current dams. If count > [threshold], default to turbine, not dam.”

Say: “Nozzle classification sun run complete. Postmortem written. Guardrail installed.”

PHASE 3 — Create the nozzle notebook

Create `work/nozzle-notebook.md`:

Nozzle Postmortem Notebook

Purpose: Preserve nozzle decisions as durable scar tissue. The notebook reveals which areas are over-dammed.

Entry Template

[DATE] — [Rule]

- **Rule:**
- **Initial classification:**
- **Operator's correction:**
- **Did it hold:**
- **If failed, pressure went to:**
- **Conversion (if applied):**
- **Postmortem lesson:**
- **Guardrail installed:**

Cluster Detection Rule

Every ten entries, count dams per area. Areas with five or more dams are over-dammed candidates. Propose conversion sweeps.

Say: "Nozzle notebook created. The cluster pattern is what to watch for."

PHASE 4 — Create the authority-boundary file

Create `work/nozzle-authority.md`:

Nozzle Authority Boundary

Purpose: Keep nozzle architecture in the operator's hands.

The Rule

- Agent classifies, proposes, inventories.
- Operator confirms, converts, removes, refines.

- No rule is added, converted, or removed without explicit per-rule approval.
- The operator decides which areas need dams.

When the Operator Must Confirm

- Any conversion from dam to turbine.
- Any new rule being added.
- Any rule being removed.
- Any inventory sweep being acted on.

Say: “Authority boundary created. The nozzle architecture belongs to the operator.”

PHASE 5 — Install the trigger

Add to operator rules:

Trigger — Nozzle Classification

When the operator says “nozzle check,” “classify this rule,” “dam inventory,” or before writing any new rule, open `work/nozzle-design-lab.md`, `work/nozzle-classification-sunrun.md`, `work/nozzle-notebook.md`, and `work/nozzle-authority.md`. Classify the rule or inventory the rule set. Surface conversion candidates. Wait for per-rule approval. Apply approved changes. Write the postmortem. Install one guardrail.

Pre-Rule Gate

Before writing any new rule, classify it (dam / turbine / channel) and surface the dam count in the surrounding area. Wait for operator decision.

Leak-Investigation Rule

When a rule fails twice, do not stack another dam. Reopen the lab, ask whether the pressure is escaping elsewhere, propose a conversion.

Say: “Trigger installed. Every new rule classified before it goes in.”

PHASE 6 — Run the first classification prompt

Ask exactly:

“What rule do you want classified? Send the rule text, or describe the rule you’re about to write, or send the operator rules file and we’ll inventory it. If you have a rule that’s failed twice, that is a strong candidate. I’ll classify, inventory the surrounding dams, propose any conversions, and wait for your decision before any change.”

Stop after asking. Wait for the operator’s rule.

PHASE 7 — Register in book-bag

Add to 🎒 book-bag.md:

💧 Nozzle-Design Lab

- **What:** Operator-confirmed dam/turbine/channel classification with conversion proposals; postmortem tracks over-dammed areas.
- **Files:** `work/nozzle-essay.md`, `work/nozzle-design-lab.md`, `work/nozzle-classification-sunrun.md`, `work/nozzle-notebook.md`, `work/nozzle-authority.md`
- **Triggers:** “nozzle check,” “classify this rule,” “dam inventory,” every new rule, every rule that has failed twice
- **Authority boundary:** Agent classifies; operator decides architecture
- **Source:** JKE University — Course 206

Say: “Nozzle lab registered. The water doesn’t stop. The nozzle is what we design.”

PHASE 8 — Write the journal entry

Add to 🏫 school.md:

[TODAY] — JKE Course 206: Nozzle-Design Lab Installed

What Shipped: Nozzle lab, classification sunrun, notebook, authority boundary, trigger, pre-rule gate, leak-investigation rule, first classification prompt. **Decisions Made:** Rules are nozzle parts. Dams cost pressure. Most rules should be turbines. Conversions beat additions. **Files Created:** `work/nozzle-design-lab.md`, `work/nozzle-classification-sunrun.md`, `work/nozzle-notebook.md`, `work/nozzle-authority.md` **Files Modified:** 🎒 book-bag.md, operator rules **Source:** JKE University — Course 206

PHASE 9 — Say exactly

“Nozzle-design lab installed. Send me a rule or your rule file. I’ll classify each as dam, turbine, or channel, inventory the surrounding dams, propose conversions where useful, and wait for your decision per rule. The water doesn’t stop; the nozzle is what we design.”

SCOPE HINT

Course 206 of 210. Level 5, Library of Instruments. Course 205 named the gradient column. Course 206 hands the agent the tool for shaping that column. Course 207 turns the lens on the platform column — what depends on the platform and what is actually yours.

END OF PROTOCOL