# Text in, model out? Measure it yourself.

Material for episode 1 of the channel «Stephan Fischli · Agentic Orchestration» · Levels 1 to 3 · as of October 2026

In the video I measured how well language models produce a BPMN model from a process description, at five levels that build on each other. With this package you can measure the first three levels with your own tool: ChatGPT, Claude, Copilot, a modelling tool with AI or whatever you use. You do not need a program, only ten runs, a BPMN viewer and this checklist.

### In the package

- `01_Notiz_Retourenprozess.txt`: the input for levels 1–3. A note written from memory about a returns process, the way such documents look in practice (English, exactly as used in the experiment).
- `02_Task_EN.txt`: the request to the AI, unchanged from the experiment (German), with a translation.
- `03_Guide_and_Checklist_EN.pdf`: this document, with the score sheet and the reference values from the video.
- `04_Score_sheet_EN.csv`: the score sheet to fill in with Excel.
- `Zusatz_Stufe4_drei_Unterlagen/`: Word document, training slides and the note. They deliberately contradict each other a little; for everyone who also wants to try level 4.
- The files without _EN are the same in German.

All documents are fictional; companies, people and processes are invented.

---

## The experiment in three steps

**1** **Ask ten times**

For every run, open a **new** conversation (no history, no project knowledge) and send exactly this message: the line from `02_Task_EN.txt`, an empty line, then the whole text from `01_Notiz_Retourenprozess.txt`. Do not answer follow-up questions, do not correct anything. What is there after the first answer counts.

**2** **Save the result**

Copy the XML from the answer into a file `run01.bpmn` to `run10.bpmn`. If the tool offers a file to download, take that one. If it delivers no XML at all, level 1 is not reached for this run.

**3** **Check and record**

Check every run against the checklist on the next page and record the result in the score sheet. At the end, count per level: how many of ten runs reach it.

> **Why ten times?** A single result can look good. Whether a tool is reliable only shows when you repeat the same question. That is exactly what level 3 checks.

## Checklist

### 1 Level 1 · It loads

- Open the file in a BPMN viewer, for example **demo.bpmn.io** (in the browser, no sign-up) or the Camunda Modeler.
- **Reached** if the file opens without an error message **and** the diagram is drawn completely: all tasks, decisions, events and arrows are visible.
- Typical errors from the experiment: the XML is broken (e.g. a `--` in a comment), or the diagram part (`bpmndi:BPMNDiagram`) is missing entirely; then the viewer shows "no diagram to display". Both count as **not reached**, even if the process is described correctly in the text.

### 2 Level 2 · Complete

- The note describes a normal case and **two exceptions**. Both must appear as a path of their own leading to an end of their own:
  - **Request rejected**: the returns team does not approve the refund.
  - **Goods not in order**: the warehouse finds that it is the wrong item or that the condition is not right.
- **Reached** if both exceptions are modelled. A run that does not reach level 1 can still be checked for level 2 (look in the text or the XML); for the overall result, however, only what also loads counts.
- Not required: a deadline for the return. It is only in the additional documents, not in the note.

### 3 Level 3 · Stable

- Compare the ten runs with each other. Two runs have **the same flow** if they have the same steps in the same order and the same decisions with the same outcomes. Differences in names ("Check" or "Inspect") and in the layout do **not** count.
- Give each new flow a letter (A, B, C ...) and record it in the score sheet.
- The **result** is the number of different flows. Ideal is 1: asked ten times, ten times the same process.

> **Levels 4 and 5 (for the curious).** Instead of the note, take the three documents from the additional folder. The note says: refund only, no formal closure, no supervisor, and internally it is called "Refund Request". The slides and the Word document partly say the opposite.
> **Level 4 · Faithful**: does the model silently adopt one side, or does it invent something? Does the term "Refund Request" stay?
> **Level 5 · Honest**: does the tool ask before it decides, or does it mark its assumptions visibly in the model?

## Reference values from the video

Input: the note · 10 runs each · one message without a system prompt · default temperature · called via the API, October 2026. Evaluated automatically; when checking by hand, individual runs may be classified differently.

| Tool / model | Level 1<br>loads with diagram | Level 2<br>path "goods not in order" | Level 3<br>different flows |
|---|---|---|---|
| gpt-oss (120b) | 0 / 10 | 3 / 10 | 6 |
| DeepSeek V4 Flash | 0 / 10 | 1 / 10 | 5 |
| GPT-4.1 | 0 / 10 | 4 / 10 | 7 |
| GPT-5 mini | 3 / 10 | 8 / 10 | 7 |
| GLM-5 | 10 / 10 | 1 / 10 | 6 |
| **Lab with method (gpt-oss)** | **10 / 10** | **9–10 / 10** | **5–7** |

All models had the path "request rejected" in all runs. Values regardless of whether the run also reaches level 1. "Lab with method": the same model gpt-oss, embedded in a method (fixed schema, style rules from "BPMN Method and Style", layout convention, check before output); implemented in my lab tool MESTRO (ELCA).

## Score sheet

Tool / model: _____ Date: _____

| Run | Level 1<br>loads with diagram | Level 2<br>"rejected" | Level 2<br>"not in order" | Level 3<br>flow (A, B …) | Remark |
|---|---|---|---|---|---|
| 1 | | | | | |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |
| 5 | | | | | |
| 6 | | | | | |
| 7 | | | | | |
| 8 | | | | | |
| 9 | | | | | |
| 10 | | | | | |
| **Total** | **__ / 10** | **__ / 10** | **__ / 10** | **__ flows** | |

> **Your level:** level 1 counts as reached if all ten runs load; level 2 if, in addition, all ten contain both exceptions; level 3 if, in addition, all ten have the same flow. Tell me in the comments under the video which tool reaches which level, and where it breaks for you.

bpmn.ch · Stephan Fischli · Material for episode 1 "Text in, model out?". Free to use for your own measurements and training, with attribution.