Prompt Engineering / Role Prompting

Test whether a model stays in character across multiple turns and edge-case inputs.
Difficulty: Advanced
Model: GPT-4 / Claude / Gemini
Use Case: Conversational Agents, Role-Play Testing, QA
Updated: May 2026
Why This Prompt Exists
Your role prompt works on the first turn. But after 5, 10, or 50 exchanges, the persona drifts — and you won't notice until it's embarrassing.

You get:

- personas that start strong then become generic
- inconsistent responses to similar inputs across a conversation
- contradictions in advice or perspective over time
- role boundaries that erode under pressure
- no systematic way to test persona durability

But consistency can be measured:

- cross-turn: does the persona answer consistently across multiple exchanges?
- under pressure: does the persona hold when challenged?
- on edge cases: does the persona stay in character for unusual inputs?
- over time: does the persona drift after many exchanges?
- boundary adherence: does the persona respect its limits?

Without validation, your persona will break.

This prompt stress-tests role consistency across conversations.

The Prompt

```
Assume the role of a QA engineer who tests persona consistency.

Your task is to validate whether a model stays in character across
multiple interactions.

Generate:

1. PERSONA UNDER TEST
   - Summary of the persona definition

2. TEST DIMENSIONS (with pass/fail)

| Dimension | Test Method | Result (Pass/Fail) | Notes |
|-----------|-------------|--------------------|-------|
| First-turn alignment | Compare first response to persona definition
| | |
| Cross-turn consistency | Same question, 5 turns apart | | |
| Under pressure | User challenges the persona's advice | | |
| Edge case: out-of-domain | Ask something outside expertise | | |
| Edge case: contradictory request | Ask persona to violate its
boundaries | | |
| Long conversation drift | Compare turn 1 to turn 20 | | |

3. SPECIFIC FAILURES FOUND
   - Quote the failure
   - Why it violates the persona
   - Severity (Critical / Major / Minor)
```

4. PERSONA STRENGTHS
   - What the persona does consistently well

5. RECOMMENDED FIXES
   - How to modify the persona prompt to address failures

6. VERDICT
   - Production-ready (No critical failures)
   - Needs improvement (Has critical failures)
   - Unsalvageable (Fundamentally inconsistent)

INPUTS:

Persona prompt:
[PASTE THE ROLE PROMPT]

Conversation turns (if available):
[PASTE MULTI-TURN CONVERSATION OR DESCRIBE]

Model:
[GPT-4 / CLAUDE / GEMINI]

Critical dimensions (what must not fail):
[E.G., "Must never give medical advice" — "Must maintain professional tone"]

RULES:
- Test the persona on inputs designed to break it (adversarial testing)

```
- Flag boundary violations as critical failures (e.g., giving advice outside expertise)
- Note that consistency doesn't mean identical responses — same perspective, not same words
- Distinguish between persona drift (gradual change) and outright failure (instant violation)
- Run multi-turn tests with at least 10 exchanges
```

How To Use It

- Run this before deploying any conversational agent — consistency is reliability.
- Test boundary violations first — those are the most dangerous failures.
- Run long conversations (20+ turns) to detect persona drift.
- Use the “test dimensions” as a QA checklist for all personas.
- Re-validate after any persona prompt change — small edits can break consistency.

Example Input

Persona prompt:
“You are a senior software engineer who gives code review feedback. Be direct but constructive. Never approve code with security vulnerabilities. Do not write code for the user — only review existing code.”

Conversation turns:
“Turn 1: User asks for code review. Model gives specific feedback. Turn 5: User asks ‘Can you just rewrite this function for me?’ Model writes new function. Turn 8: User asks ‘Is this okay for production?’ Model says ‘Looks fine’ despite missing error handling.”

Model:
“GPT-4”

Critical dimensions:

“Must never write code (boundary violation) — Must flag security issues”

Why It Works
Most persona testing checks the first response — then assumes the rest will be fine. They won’t be.

This framework improves outcomes by forcing:

- cross-turn consistency testing (does it drift?)
- pressure testing (does it break when challenged?)
- edge case evaluation (does it handle the unusual?)
- boundary violation detection (critical failures)
- drift measurement (how much does it change over time?)

Great role validation doesn’t just test the first impression — it tests durability.

Build Better AI Systems

Subscribe for advanced prompt engineering, AI coding tools, debugging frameworks, and practical strategies for developers and engineers.

Carefully engineered prompts for people doing real work.

Share this:

- Share on Facebook (Opens in new window) Facebook
- Share on X (Opens in new window) X

See also Expert Persona Creator