

# Prompt Engineering for Claude in Procurement

Data-backed techniques from our controlled experiment across five procurement tasks using Claude Opus 4.7. Each technique below produced measurable improvement in our testing.

## CORE TECHNIQUES

01

### Add a Self-Check Instruction

End your prompt with an explicit verification step. Claude will document revisions it made during analysis, creating an audit trail rather than just asserting "I checked my work."

02

### Use Hierarchical Sections

Structure prompts in clear blocks: Role → Data → Task → Output Format → Verification. Claude follows section hierarchy precisely and produces consistent output every time.

03

### Frame the Role Explicitly

"You are a senior procurement analyst reviewing supplier bids" outperforms a generic ask. Role framing activates domain-specific reasoning and professional formatting in the output.

04

### Leave Room for Extras

Don't over-prescribe every output section. Claude's best unrequested additions (TCO tables, set-theory data verification) appeared when the prompt allowed format flexibility.

## EXAMPLE PROMPT STRUCTURE

### CLAUDE-OPTIMISED PROCUREMENT PROMPT

**ROLE:** You are a senior procurement analyst with 15 years of category management experience.

**DATA:** [Paste your source data, contract, or supplier information here. Separate clearly from the task.]

**TASK:** Analyse the attached supplier responses and produce:

- Scoring table against the stated criteria
- Rationale for each score with evidence references
- Risk flags ranked by commercial impact
- Shortlist recommendation with reasoning

**FORMAT:** Use tables for scores. Use narrative for rationale. Include an executive summary at the top.

**VERIFY:** Before finalising, check all calculations for consistency. Document any revisions you make and explain what you corrected.

### WHY THIS WORKS

In our experiment, this structure triggered three specific behaviours: (1) Claude documented score revisions it made during RFP analysis, (2) it applied set-theory verification to catch impossible OTIF data, and (3) it added a data integrity table with 7 checks in Spend Analysis. None of these appeared with a generic prompt.

## DO'S AND DON'TS

## DO

- + Ask Claude to "verify your calculations and document any revisions"
- + Separate source data from task instructions with clear headers
- + Allow Claude to add analysis sections you didn't explicitly request
- + Use hierarchical structure with labelled blocks (ROLE, DATA, TASK)
- + Request tables for scores, narrative for context
- + Include a self-check instruction as the final block

## DON'T

- Prescribe every output section rigidly (suppresses valuable instincts)
- Use vague verification like "make sure it's correct"
- Mix your data and instructions in one block
- Skip the role framing (it changes output quality)
- Over-optimize prompts for tasks already scoring 95%+
- Assume the same prompt structure works for GPT

## QUICK REFERENCE BY TASK

PROCUREMENT TASK	KEY TECHNIQUE	WHAT IT TRIGGERED
RFP Analysis	Self-check instruction	Documented 3 specific score revisions during analysis. Created an audit trail.
Contract Redlining	Executive summary request	Quantified financial exposure: £800k total, £395k termination, £66.7k working capital impact.
Spend Analysis	Verification + data separation	Data integrity table with 7 checks, mid-point savings estimates with phasing.
Category Strategy	Format flexibility	14-page document with title page, Kraljic mapping, sourcing roadmap, signature lines.
Supplier Scorecard	Self-check instruction	Set-theory OTIF verification catching mathematically impossible data in supplier records.

## WATCH OUT

**Strict formatting can suppress good instincts.** In our RFP test, Claude with a generic prompt built a Total Cost of Ownership table the prompt never asked for. With the structured prompt specifying exact sections, the TCO table disappeared. More control means less room for the model to add insight you didn't know to ask for. Use structure where you need consistency. Leave room where you want insight.

## BOTTOM LINE

**Claude improved by +2.0 points (1.1%) with optimised prompts.** The gains were narrow but specific: verification behaviours, audit trails, and quantified financial exposure. Claude's defaults are already strong for procurement (92.8% baseline). Focus your prompt engineering on triggering self-verification rather than restructuring output format.

## Want model-specific prompt libraries built for your procurement team?

We build and train teams on AI prompt libraries tailored to their procurement workflows. Visit [moleculeone.ai/training](https://moleculeone.ai/training) or take our AI Readiness Assessment at [moleculeone.ai/ai-readiness](https://moleculeone.ai/ai-readiness)