From Basics to Brilliance:

A Quality-First Guide to Prompt Engineering

Ten10

Visit **ten10.com**

Contents

Written by

Stuart Day
Ten10 Managing Principal Consultant

James Stubbs
Ten10 Principal Consultant

Jorge Millares-Bobet
Ten10 Principal Consultant

Introduction

In their race to adopt Large Language Models (LLMs) and increase efficiency, many organisations encounter a fundamental paradox: LLMs are, by design, creative and probabilistic. Asking the same question twice often yields two different answers—a "feature" for a poet, but a liability for a business process.

The problem facing modern industry is the "Slot Machine" approach to AI—inputting vague requests and hoping for a usable result. To transform AI from a generic chatbot into a reliable pipeline utility, organisations must move beyond simple "asking" and embrace the science of Prompt Engineering: the art of constraining AI to deliver deterministic, repeatable, and structured outputs every time.

Read on to learn our structured steps to take you on that journey. It starts with a shift in your mindset:

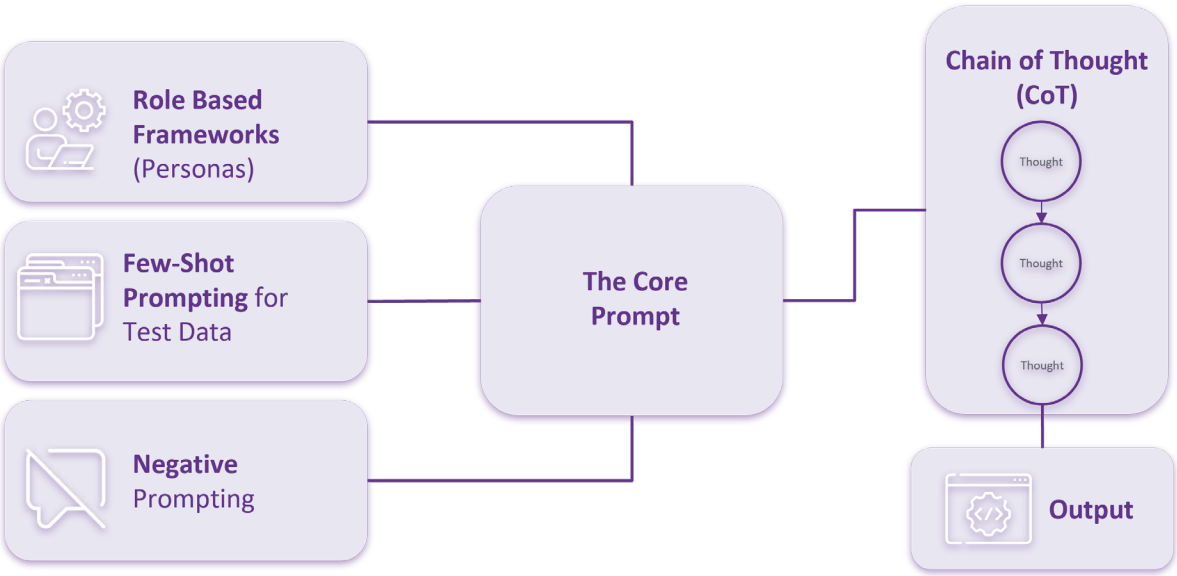
Step 1: Shift your mindset

The journey from basics to brilliance begins with a change in perspective. Stop treating AI like a search engine where keywords suffice. Instead, treat it like a highly capable but very literal **junior engineer**.

A search engine requires a query; a junior engineer requires a **brief**.

Brilliance is achieved when you provide unambiguous instructions that ground the AI in your reality rather than the "average" of its training data.

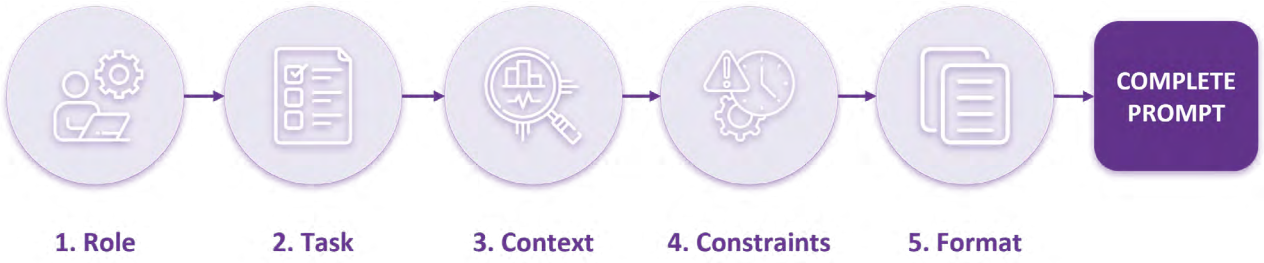
Step 2: Establish the foundations



Before building a complex prompt, you must understand the core techniques that guide an AI's reasoning:

- **Role-Based Personas:** Restrict the AI's knowledge base to a specific expert field rather than the "average" of the internet.
- **Chain of Thought (CoT):** Force the model to follow a step-by-step reasoning process, which is essential for complex tasks.
- **Few-Shot Prompting:** Teach the AI by providing examples of what "good" looks like, locking in the desired pattern for the output.
- **Negative Prompting:** Use the "negative space" to define what the AI should not do, creating essential guardrails.

Step 3: Master the anatomy of a professional prompt



The journey from basic, inconsistent queries to consistently brilliant outcomes necessitates a fundamental shift in how we construct prompts. We must move away from intuition and embrace a modular, structured methodology. To move away from probabilistic chaos, every prompt should follow a five-part anatomy to ensure a "flawless brief":

- **Role (The Expert Persona):** Never let the AI act as an average internet user. Assign a professional persona—such as a "Senior QA Automation Engineer with 15 years of experience"—to force the model to pull from high-quality, authoritative training data rather than beginner-level fluff.
- **Task (The Action Engine):** Replace passive words like "help me" with precise action verbs such as Analyse, Critique, Extract, or Generate.
- **Context (Grounding in Reality):** AI does not know your proprietary business logic. "Context" is the act of "bringing the car to the garage" by pasting raw materials like Jira tickets, API Swagger docs, or JSON payloads directly into the prompt to prevent hallucinations.
- **Constraints (The Negative Space):** Define what the AI must not do. Guardrails such as "do not test the backend" or "no conversational filler" ensure the AI stays in its lane and doesn't waste its output limit on irrelevant information.
- **Format (The Last Mile):** Dictate the exact structure required—be it Gherkin syntax, a JSON array, or a Markdown table. This eliminates manual reformatting and makes the output immediately ready for your toolchain.

Quality Tip:

As with anything we build, prompts must also be objectively measurable. Try scoring every prompt against three criteria:

- **Repeatability**
- **Testability**
- **Consistency**

If the identical prompt yields divergent results across three separate attempts, it is a definitive sign that the prompt lacks sufficient structure and control, primarily in the *Instruction* and *Constraint* components.

Step 4: Move to scenario-based engineering

"Brilliance" in Prompt Engineering is achieved not through complexity, but through specialisation. It is the transition from constructing one-off, general-purpose prompts to engineering repeatable systems tailored for specific business scenarios. The structural logic of a prompt must dynamically shift based on the specific business value being sought.

Scenario	Logic Focus	Critical Components	Desired Outcome
Code Generation	Syntactic Rigour & Edge Cases	Strict Constraints, Environment Context	Production-ready, securely written code that passes unit tests.
Data Analysis	Logic-Chain & Chain of Thought	Explicit Step-by-Step Instructions, Ground Truth Context	Insightful, evidence-based conclusions with traceable steps.
Executive Reporting	Tone, Clarity & Synthesis	Strict Formatting/ Length Constraints, Audience-Specific Context	High-level, action-oriented summaries that distil complexity into simple decisions.

By meticulously tailoring the structural logic and weighting the four prompt components (Context, Task, Instruction, Constraint) to the specific scenario, you transform the LLM from a generalist assistant into a highly specialised, domain-aware tool that fits seamlessly into existing workflows.

Ten10

Step 5: Mitigate risk with active guardrails

The most significant barrier to enterprise-wide AI adoption is the risk profile, dominated by the threat of "hallucinations" and the potential for model misuse. Achieving professional-grade, trustworthy results demands the implementation of active, engineering-grade risk mitigation techniques:

Chain of Thought (CoT) & Zero-Shot CoT: This is a meta-instruction that explicitly asks the AI to "think step-by-step before providing the final answer." This technique forces the model to articulate its reasoning, process logical steps, and self-correct potential errors before committing to an output. In complex tasks, CoT can reduce factual errors and logical inconsistencies by over 40%.

Context-Driven Memory (RAG/Grounding): Never rely solely on the model's vast, but potentially outdated or non-proprietary, internal training data. To ensure accuracy and relevance, you must provide the model with "ground truth" documents, specific company data, or proprietary knowledge to reference. This Retrieval-Augmented Generation (RAG) approach anchors the output to verifiable external facts, eliminating unsupported claims.

Continuous Human Oversight (The Force Multiplier): While AI provides unprecedented speed, it is not a replacement for strategic decision-making. Rigorous quality engineering necessitates a human expert to validate and critically assess the model's outputs, particularly in high-stakes areas like regulatory compliance or financial reporting. At Ten10, we view AI as a "force multiplier" that amplifies human capability, not a substitute for human strategic expertise.

Step 6: Embrace agentic automation

The pinnacle of Prompt Engineering—where the discipline realises its full business value—is the shift toward Agentic AI. This transcends simple, chat-based question-and-answer interactions. Agentic workflows define autonomous systems in which sequences of structured prompts drive AI "agents" that can interact with existing internal tools execute commands in CI/CD pipelines, query live databases, and initiate further actions.

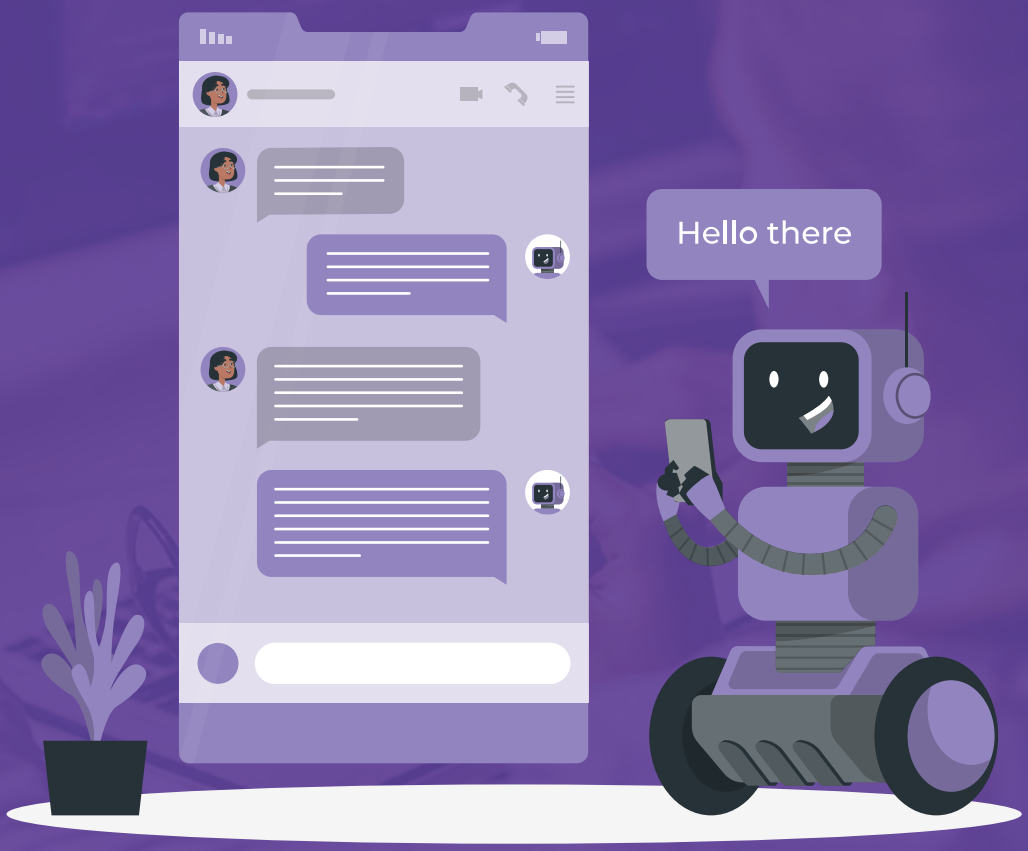
In this advanced stage, prompt engineering becomes the orchestration layer. The focus shifts from asking for a singular response to defining the comprehensive parameters for an autonomous agent to observe, plan, act, and reflect—allowing it to solve complex, multi-step business problems without constant human intervention.

Ready to go from basics to brilliance?

In the next section, you'll find 12 professional-grade prompts to help get you started. We've mapped each part of the prompt to the structure used earlier in this whitepaper, so you can adapt any section you need to make them even more effective while using them in your workflows.

Ten10

12 Professional-Grade Quality Engineering Prompts



1. The Bug Triage Summariser

```
Act as a Senior QA Lead.

Extract the critical defects, priority shifts, and immediate
blockers from the provided triage meeting notes.

[Insert Raw Bug Triage Transcript/Notes]

Do not include minor UI polish discussions or “won’t fix” items.
Limit the summary to one page.

Output a Markdown table with three columns: Bug ID/Feature,
Severity, and Assigned Developer.
```

2. The Defect Report Diplomat

```
Act as a Senior Quality Engineer with 15 years of experience in
cross-functional communication.

Generate a professional, objective response to a developer who
has marked a critical bug as “Working as Intended” or “Can’t
Reproduce.”

[Insert Developer’s Comment and Original Bug Report]

Do not use accusatory language. Do not demand an immediate fix.
Focus strictly on the user impact and the steps to reproduce.Limit
the summary to one page.

Format as a three-paragraph email and Jira comment draft ready to
be posted.
```

3. The Flaky Test Analyser

```
Act as a Lead SDET (Software Development Engineer in Test).

Analyse the provided CI/CD pipeline execution logs and identify
three core patterns causing test instability.

[Insert Raw Logs or Paste Test Execution CSV]

Think step-by-step before providing the final answer. Distinguish
between environment timeouts and actual logic regressions. Do not
guess root causes not supported by the logs.

Output a bulleted list with a brief executive summary of the
“Stability Score” at the top.
```

4. The Automation Framework SWOT Strategist

```
Act as a Head of Quality Assurance.

Critique the feasibility of adopting a new testing tool or
framework based on the team’s current technical stack and project
goals.

[Insert Tool Documentation / Team Tech Stack / Project
Requirements]

Use the negative space to define what not to do: do not suggest
tools that require languages the team doesn’t currently use.

Format as a structured four-quadrant SWOT (Strengths, Weaknesses,
Opportunities, Threats) analysis of the tool’s integration.
```

5. The User Story Refiner

```
Act as a Quality Advocate and Business Analyst.

Synthesise this dense Product Requirement Document (PRD) into
simple, testable Acceptance Criteria for the engineering team.

[Insert Dense Feature Requirement Document]

Maintain all functional edge cases. Avoid vague terms like “user-
friendly” or “fast.” Ensure every point is binary (Pass/Fail).

Output a numbered, step-by-step list of Acceptance Criteria
(Given/When/Then format preferred).
```

6. The “Test Coverage” Pitch Customiser

```
Act as a Head of Quality Assurance.

Adapt a general proposal for “Increased Automation Coverage” to
directly address the specific pain points of a Product Manager
concerned about release velocity.

General Proposal: [Insert Automation ROI Pitch] AND PM’s Concerns:
[Insert Recent Sprint Delay Notes / Feedback]

Maximum 250 words. Do not focus on technical debt; focus on “speed
to market” and “risk mitigation.”

Format as two short paragraphs followed by a single closing
question designed to secure a dedicated “Quality Sprint.”
```

Ten10

7. The Inclusive Persona Generator

```
Act as a Senior Accessibility Tester.

Generate a set of inclusive user personas to be used for edge-case testing, ensuring diverse physical and cognitive needs are represented.

[Insert Feature Description and Target Audience]

Avoid stereotypes. Include specific assistive technologies (e.g., screen readers, switch access). Do not exceed 500 words.

Output a structured document divided into: Persona Profile, Primary Goal, Technical Constraint, and Suggested Test Scenarios.
```

8. The Test Cycle Post-Mortem Drafter

```
Act as a QA Release Manager.

Extract the key testing successes and delivery bottlenecks from our recent release cycle feedback.

[Insert QA Team Feedback / Slack Release Channel Logs]

Maintain the anonymity of all testers and developers. Focus strictly on the environment's stability and testing windows, not individual performance.

Format as a "Start, Stop, Continue" bulleted list regarding the QA process.
```

9. The Exploratory Testing Ideator

```
Act as a Creative Quality Engineer.

Generate five “outside-the-box” exploratory testing hooks/scenarios
to break a new feature.

[Insert Feature Functional Specification]

No “Happy Path” scenarios. Focus on concurrency, interrupted
states, and data boundary violations. Keep the tone professional.

Output a numbered list of test ideas, each including a “What if?”
statement and the expected system failure.
```

10. The Environment Setup SOP Documenter

```
Act as a Senior SDET / DevOps Engineer.

Analyse the provided rough notes on local environment configuration
and document a clear Standard Operating Procedure for new hires.

[Insert Rough CLI Commands / Dependency Notes / Setup Steps]

Force a step-by-step reasoning process to ensure no hidden
dependencies are missed. Use “Warning” callouts for common
“gotchas” in the installation process.

Format as a clear instructional document with headers and bolded
cautionary callouts.
```

11. The Data Trend Analyser

```
Act as a Lead Data Analyst.

Analyse the provided raw performance metrics and identify three
core trends.

[Insert Raw Data or Paste CSV Text]

Think step-by-step before providing the final answer. Rely solely
on the provided ground truth context and do not hallucinate
missing data.

Output a bulleted list with a brief executive summary at the top.
```

12. The Project Post-Mortem Drafter

```
Act as an Agile Scrum Master.

Extract the key successes and bottlenecks from our recent project
feedback notes.

[Insert Team Feedback Notes / Sprint Retro Board Text]

Maintain the anonymity of all team members. Focus strictly on
processes, not people.

Output as a “Start, Stop, Continue” bulleted list.
```

Prompt engineering is the critical bridge between the raw potential of Large Language Models and the tangible, measurable business value they must deliver.

By adopting a quality-driven discipline—one deeply rooted in structured context, scenario-specific logic, and proactive risk mitigation—organisations can move definitively beyond the basics.

This strategic, quality-engineering-led approach is the only path to unlocking the true brilliance and sustained confidence of your AI investments.

If you're starting a major transformation or automation project, get in touch with our team to learn more about the AI, Intelligent Automation, and Quality Engineering consultancy services we offer to accelerate and support you on your journey.

P: +44 (0)207 100 7794

E: contact@ten10.com

W: ten10.com

Ten10