

BEYOND BETTER PROMPTS

Better Prompts Will Not Fix Undecided Requirements

Why generative AI turns organisational ambiguity into system behaviour

Giovanni Leonardi

May 2024



Better Prompts Will Not Fix Undecided Requirements

Transformation · No. 314

Originally written May 2024 | Re-edited and re-rendered for the WhereBeyond Edition, 2025

*“Generative AI has made the interface conversational.
It has not made organisations coherent.”*

1. The Specification Hidden in the Prompt

At 9.10 on a Monday morning, a transformation team demonstrates a new generative assistant to the operations director. The assistant has been given a twelve-line prompt and a folder of policy documents. It produces a plausible response in seconds. The room is impressed. By Wednesday, the same assistant has invented an approval step, overlooked an exception for high-value cases, and answered the same question differently when the user changes a single adjective.

The immediate reaction is familiar: improve the prompt.

A specialist expands twelve lines to forty-seven. Examples are added. The desired tone is defined. Prohibited answers are listed. A paragraph explains when to abstain. Another identifies the sources that should take precedence. The output improves, but the exercise has revealed something more important than a prompting technique. The team is doing requirements engineering without calling it requirements engineering.

That is the pattern emerging in 2024. Prompt engineering is often presented as a new craft concerned with eliciting better behaviour from large language models. At its most serious, however, it is an old organisational discipline returning in unfamiliar clothes: the discipline of making intent explicit enough that a system can act on it. The novelty lies in the interface. The underlying difficulty remains the same. Organisations do not struggle primarily because they lack the right words for a machine. They struggle because they have not agreed what they want, which evidence should govern, where judgement is permitted, and who carries the consequence when the answer is wrong.

2. Natural Language Does Not Remove the Need to Specify

The attraction of the prompt is its informality. A manager can type an instruction in ordinary language and receive an apparently finished result. There is no long development cycle between desire and response. This creates the impression that specification has disappeared.

It has not disappeared. It has moved closer to the user and become easier to overlook.

Traditional requirements work made ambiguity visible through documents, workshops, process maps, acceptance criteria and change requests. These devices could become ponderous, but they forced disagreement into the open. A conversational interface conceals disagreement because the system will usually produce something. Fluency arrives before correctness. The answer reads smoothly enough to postpone the question of whether the organisation ever defined “correct”.

Consider the apparently simple instruction: review this supplier proposal and identify the principal risks. Before a reliable result is possible, someone must settle a chain of requirements:

- What counts as a principal risk: financial exposure, delivery uncertainty, regulatory breach, reputational harm, or some combination?
- Which sources carry authority when the proposal conflicts with internal policy?
- Should absence of evidence be reported as a risk, treated as neutral, or trigger a request for clarification?
- Are risks ranked by probability, impact, detectability, or the executive appetite of the moment?
- Which conclusions may the system draw, and which must be referred to a commercial, legal or operational specialist?

These are not linguistic decorations. They are decisions about purpose, evidence, boundaries and accountability. A better adjective will not settle them.

A prompt is not merely an instruction to a model. It is a compressed operating agreement between the organisation, its information and the person expected to rely on the result.

3. Why the Prompt Became the Container for Unfinished Decisions

The persistence of “better prompting” as the default remedy is not accidental. It is sustained by several structural forces visible across generative AI programmes.

First, the technology invites local experimentation. A useful prototype can be assembled by one capable person with access to a model, a small collection of documents and a few hours. This is a strength: it lowers the cost of learning. But the prototype also becomes the place where policy decisions accumulate silently. Each time an awkward case appears, another line is added to the prompt. Soon the prompt contains fragments of operating procedure, quality policy, risk appetite and editorial guidance, with no clear owner for any of them.

Second, prompts look inexpensive to change. A paragraph can be rewritten without a release process or a formal change request. That apparent ease disguises the cost of interaction. One instruction may alter tone, completeness, source selection and willingness to refuse all at once. The prompt is changed locally; the consequences spread across the behaviour of the system.

Third, the quality problem is mistaken for a wording problem because language is the visible surface. When an answer fails, the failed sentence is what people see. The hidden causes may be contradictory source material, missing context, an unsuitable task, weak retrieval, or an unresolved business rule. Editing the prompt is attractive because it is immediate and demonstrable, even when it is not causal.

Fourth, ownership is convenient but misplaced. The person who built the demonstration becomes the “prompt expert”, and therefore the recipient of questions that belong to process owners, risk specialists and information stewards. A technical craft absorbs decisions the organisation has avoided making elsewhere.

The result is a curious inversion. We celebrate the accessibility of natural language while creating important operational specifications that only one or two specialists understand well enough to alter.

4. A Forty-Seven-Line Prompt and the Missing Rule

A composite example shows how quickly this happens. A service organisation pilots an assistant to classify incoming customer correspondence and draft a proposed reply. The weekly volume is 8,000 items. In the pilot sample of 400, the first prompt assigns the correct category in 83 per cent of cases. The team sets a target of 92 per cent before wider use.

Over three weeks, the prompt grows from 14 lines to 47. It contains six examples, nine “always” instructions and seven “never” instructions. Classification reaches 91 per cent. Yet one category—requests involving a disputed charge and a vulnerable customer—remains unreliable. The assistant alternates between the complaints route and the hardship route.

The prompt specialist tries clearer definitions. The operations manager supplies more examples. The model continues to vary because the underlying procedure itself is unresolved: the two teams use different routing rules, and neither rule specifies which condition takes precedence when both apply.

The turning point is not a clever prompt. It is a ninety-minute decision session involving the process owner, the two team leaders, a risk representative and the pilot lead. They agree that vulnerability takes precedence, that disputed-charge evidence must travel with the case, and that no reply should be drafted until a trained reviewer confirms the route. Those decisions are expressed as three acceptance tests and then reflected in the prompt and workflow. On the next 400-item set, category accuracy reaches 94 per cent; more importantly, all 18 dual-condition cases follow the agreed route.

The figure matters less than the mechanism. Performance improved when an organisational ambiguity became a governed rule. The prompt was the last mile of the decision, not its source.

5. The Strong Case Against Formalising Too Soon

There is a serious objection to treating prompts as requirements. Generative systems are probabilistic, use cases are still being discovered, and the fastest route to value is often experimentation. Importing the full apparatus of conventional requirements management could suffocate learning. If every prompt adjustment demands a committee, a document and a sign-off, teams will preserve old bureaucracy while losing the chief advantage of the new technology: rapid dialogue between an idea and its result.

This objection is right about the danger. Early generative AI work needs room for play. Many requirements cannot be known before people encounter actual outputs. Examples frequently teach more than abstract specifications. A prompt should not be frozen while the task itself is still being understood.

But the choice is not between improvisation and bureaucracy. The more useful distinction is between exploratory prompts and operational prompts.

State	Primary purpose	Appropriate discipline	Evidence of readiness
Exploratory	Discover whether a task is useful and tractable	Rapid iteration, example collection, explicit assumptions	Repeated value in representative cases
Transitional	Define intended behaviour and expose exceptions	Named owner, test set, source hierarchy, failure review	Stable rules and understood failure modes
Operational	Support a recurring decision or action	Version control, acceptance thresholds, monitoring, change authority	Accountable use within agreed boundaries

Confusion arises when an exploratory artefact is carried into operational use without passing through the middle state. The demonstration works often enough to attract sponsorship; access expands; users assume consistency; and the prompt remains governed as if it were still a private experiment. Formality is not required everywhere. It is required at the point where other people begin to rely on the result.

6. Requirements for Behaviour, Not for a Single Answer

Treating prompt engineering as requirements engineering does not mean pretending a generative model is deterministic. It means specifying the conditions under which variable answers remain acceptable.

That changes the unit of thought. The requirement is rarely “produce exactly this sentence”. It is more often a behavioural envelope:

- Purpose: the decision or task the output is intended to support.
- Context: the information the model must receive and the circumstances the user must declare.
- Authority: the sources that govern, including precedence when sources conflict.

- Constraints: what the system must not infer, disclose, recommend or omit.
- Escalation: the conditions under which it must abstain or refer the case.
- Evaluation: representative examples, edge cases and thresholds by which behaviour will be judged.
- Accountability: the role that approves the rule and the role that accepts or rejects the output in use.

This envelope recognises variation rather than wishing it away. It also separates distinct failure classes. If the answer lacks a relevant fact, the retrieval or source set may be at fault. If it follows the wrong policy, the authority hierarchy may be unclear. If it sounds convincing while exceeding its mandate, the boundary and escalation requirements are weak. If different reviewers disagree about whether it is good, the evaluation criteria are unfinished.

The practical consequence is that prompts should be reviewed alongside examples and failure records, not in isolation. A forty-line prompt can look comprehensive while failing every difficult case. A ten-line prompt can be adequate when it sits within a well-designed workflow, receives reliable context and hands uncertain cases to an accountable person.

7. The Prompt as an Organisational Mirror

Requirements work has always exposed more than system needs. It reveals disagreements about how the organisation believes it operates. Generative AI intensifies that effect because it reaches into work previously protected by human interpretation: summarising evidence, drafting judgement, choosing emphasis, classifying ambiguous requests and proposing action.

When a team cannot write a stable instruction, the problem may not be a lack of prompting skill. It may be that policy contains contradictions, expertise is tacit, exceptions have multiplied, or authority has never been assigned. The model becomes an uncomfortable mirror. It asks the organisation to state a rule where experienced people have relied on context, relationships and discretion.

Some of that discretion should remain human. The purpose of specification is not to convert every judgement into a rule. It is to distinguish three things that are too often blurred:

1. Decisions that can be expressed and tested as rules.
2. Judgements that can be supported by generated analysis but must remain with an accountable person.
3. Situations too ambiguous, sensitive or consequential for the system to enter at all.

That distinction is itself a requirement. Without it, “human in the loop” becomes a reassuring phrase rather than an operating design. A reviewer presented with hundreds of fluent drafts may provide little real control. A reviewer given defined escalation cases, visible sources and authority to reject the output performs a different role entirely.

8. What This Changes for Transformation Leaders

The prompt should cease to be treated as the private property of the person who can make the model behave. Its language may be maintained by a specialist, but its important decisions belong to the business.

This does not require reviving vast requirements catalogues. A proportionate discipline can be compact. For each use case moving beyond exploration, leaders should expect a named purpose, an accountable process owner, a defined source set, a small but representative test pack, explicit refusal or escalation conditions, and a record of material changes. These artefacts are modest. Their value lies in forcing the right conversations before convenience becomes dependence.

The deeper implication concerns the shape of AI transformation. Many organisations are searching for scarce prompt engineering talent as though better incantations will bridge the distance between a general model and reliable work. Some specialist skill is plainly valuable. Yet the enduring capability will be broader: people who can translate between operating intent, information, risk, user behaviour and model behaviour.

We already know this work, though we may not recognise it in its new form. It is the patient work of clarifying purpose, resolving exceptions, testing assumptions and assigning authority. Generative AI has made the interface conversational. It has not made organisations coherent.

The most important prompt is therefore not the one typed into the model. It is the question put back to the organisation: what, precisely, have we decided?