

BEYOND THE DEMO

The Capability Cliff: Why Automation Fails When Organisations Do Not Understand the Work

What the rush from generative AI pilots to autonomous delivery reveals about transformation reality

Giovanni Leonardi

March 2025



The Capability Cliff: Why Automation Fails When Organisations Do Not Understand the Work

Transformation · No. 403
March 2025

“Automation does not remove ambiguity; it converts ambiguity into executable behaviour.”

1. The morning after the demonstration

The demonstration lasted twelve minutes. A service request arrived in an inbox, a language model classified it, retrieved a policy, drafted a response and proposed the next action. The hand-offs that normally consumed two working days appeared to collapse into seconds. Around the table, the questions were about scale: how quickly the approach could be extended, how many people it might release, and when an autonomous version could be put into production.

The operations manager asked a less exciting question: what happened when the policy did not agree with the customer record?

There was no answer in the demonstration. There was no answer in the process documentation either.

That small silence captures a pattern emerging across transformation portfolios in early 2025. Organisations have become able to automate the visible path through work faster than they can explain the work itself. Generative AI has intensified the effect because it can produce an impressively coherent response even when the underlying operating model is incoherent. The demonstration looks complete precisely because the machine is good at concealing the gaps that the organisation has learned to live around.

The result is a capability cliff. Progress is rapid while the work follows the happy path. Then the automation encounters the tacit judgement, contradictory data, disputed ownership and accumulated exceptions on which the real operation depends. The apparent acceleration stops abruptly. What looked like a technology problem becomes an organisational comprehension problem.

2. Automation has become easier than explanation

Traditional automation imposed a useful discipline. A workflow engine demanded explicit rules. A process redesign required someone to define states, hand-offs and controls. The

work could still be badly understood, but the technology exposed that weakness early because it refused to proceed without an instruction.

Large language models alter that bargain. They can interpret variable language, infer intent and generate plausible outputs across incomplete inputs. Retrieval techniques can place policy material in context. Tool connections can allow a model to initiate actions across several systems. The emerging discussion about AI agents therefore carries a credible promise: a greater share of knowledge work may be coordinated without every branch being hard-coded in advance.

That promise is not imaginary. The strongest case for moving quickly is serious. Organisations have spent years documenting processes that changed before the diagrams were approved. Employees routinely perform work through judgement rather than formal procedure. If automation waits for perfect understanding, it will wait forever. A capable model can also reveal patterns through use: recurring exceptions, missing information and inconsistent decisions become more visible once interactions are captured. Experimentation may therefore be a method of discovery, not merely the final stage after discovery.

But this argument changes meaning when translated into a delivery target. Learning through a bounded pilot is one thing; transferring decision authority to an automated workflow is another. The former can tolerate uncertainty because its purpose is to expose it. The latter turns uncertainty into action.

Automation does not remove ambiguity; it converts ambiguity into executable behaviour.

That conversion is the mechanism behind the cliff. A person who sees two conflicting records may pause, call a colleague, search an old email or quietly apply a local convention. An automated system will do what its instructions, context and permissions make most probable. When the organisation has not decided which source is authoritative, the automation does not resolve the governance question. It resolves it accidentally, one transaction at a time.

3. The organisation knows less than its people know

The phrase organisational knowledge is often used as though knowledge were an asset held centrally. In practice, much of it is distributed among people who compensate for weaknesses elsewhere.

A policy analyst knows that one paragraph has not been applied literally since a control review three years earlier. A team leader knows which reason code is used when none of the official codes fits. A finance colleague knows that a reconciliation is formally monthly but is corrected every Thursday because the source feed arrives late. None of these accommodations is necessarily visible in the process map. Together, they make the operation work.

Consider a composite service operation preparing to automate customer refunds in the first quarter of 2025. The approved process appeared simple: verify eligibility, calculate the amount, secure approval and issue payment. The pilot team expected four system connections and three decision rules.

Observation of actual work revealed eleven queues, twenty-seven reason codes and five unofficial spreadsheets. In a sample of 600 cases, 43 per cent departed from the documented path. Of those exceptions, roughly half were not unusual customer circumstances at all. They were corrections for missing reference data, conflicting dates or thresholds that different teams interpreted differently.

The pivotal discovery was not that the process was complex. It was that the complexity came from three different sources that had been treated as one:

- Legitimate variety: circumstances in which professional judgement was genuinely required.
- Design debt: exceptions created by old product rules, duplicate controls or fragmented systems.
- Unmade decisions: cases in which two functions had never agreed who held authority.

Automating all three as exception handling would have preserved the mess and made it faster. Eliminating all three in the name of standardisation would have removed judgement that customers genuinely needed. The work first had to be understood well enough to tell the difference.

The capability cliff is reached when the machine needs an organisational decision that the organisation has been disguising as individual judgement.

4. The cliff has four edges

It is tempting to describe the problem as poor process documentation. That is too shallow. Documentation can record an operating model, but it cannot substitute for one. The recurring gaps sit across four connected capabilities.

4.1 Process truth

The formal process describes what should happen. Transaction data shows what did happen. Practitioners explain why the two differ. None is sufficient alone. Process truth emerges only when these three accounts are reconciled.

A workshop that asks subject-matter experts to validate a diagram usually captures the official story. Direct observation and case sampling reveal the workarounds. Data analysis shows their frequency and concentration. The mechanism matters: without frequency, every memorable exception appears equally important; without observation, every system code appears self-explanatory.

4.2 Decision clarity

Many knowledge processes are not sequences of tasks but chains of decisions. Eligibility, risk, priority, materiality and escalation all require criteria and authority. If a decision has

no named owner, automation teams tend to express the current habit as a rule or prompt. Habit then acquires the force of policy without ever being approved as policy.

Decision clarity requires more than a responsible individual. It requires an explicit boundary: what the machine may decide, what it may recommend, what a person must approve, and what must stop the process entirely.

4.3 Information fitness

The generative layer attracts attention, but operational reliability is usually constrained lower down. Which record is authoritative? How current must it be? What happens when identifiers do not match? Can the evidence supporting a decision be reconstructed?

A polished response generated from weak records is not an intelligence gain. It is a confidence gain without a corresponding increase in truth. Early model evaluations often focus on the quality of language or the accuracy of classification. Production readiness depends equally on lineage, freshness, access control and the ability to detect absence rather than merely interpret presence.

4.4 Learning ownership

An automated process changes after deployment, even when the model itself is unchanged. Policies move, customer behaviour shifts, exception patterns migrate and staff begin to rely on outputs differently. Someone must own the learning loop: reviewing overrides, investigating concentrations of failure, updating instructions and deciding when the automation should be constrained.

When this ownership is left with the project team, it decays after go-live. When it is left solely with technology, operational judgement is separated from technical change. Sustainable automation needs a joint operating responsibility, not a temporary handover.

Capability	Superficial evidence	Evidence that survives production
Process truth	Approved process map	Map reconciled with cases, data and observed work
Decision clarity	Rules captured in a prompt	Authority, thresholds and escalation rights agreed
Information fitness	Relevant data connected	Sources, quality limits and missing-data behaviour defined
Learning ownership	Pilot evaluation completed	Named owners review overrides, drift and control performance

5. Why transformation portfolios keep stepping over the edge

The capability cliff persists because the incentives of transformation reward visible automation earlier than organisational understanding.

A demonstration produces an object that leaders can see. A clarified decision right produces a disagreement that leaders must resolve. A connected model appears to create momentum; the discovery that two functions use different definitions appears to slow it down. Delivery governance therefore favours the artefact that signals progress and treats the work that makes it reliable as preparatory overhead.

Business cases reinforce the distortion. Benefits are often expressed as hours removed from tasks. The cost of unresolved exceptions is harder to model because it sits across rework, control effort, customer harm and management attention. A portfolio can therefore approve the saving from automating the nominal process while omitting the cost of operating the actual one.

There is also a subtler attraction. Automation permits an organisation to imagine that difficult operating-model choices can be delegated to technology. If an agent can route work dynamically, perhaps ownership need not be simplified. If a model can interpret inconsistent records, perhaps data definitions need not be reconciled. If generated guidance can adapt to each case, perhaps policy ambiguity need not be confronted.

For a time, flexibility can disguise incoherence. Then volume turns each unresolved choice into a repeated decision, and the machine repeats it with greater speed and consistency than any individual ever could. Scale does not cure the ambiguity. It industrialises it.

6. Understanding does not mean waiting for perfection

The strongest objection remains: this standard risks becoming a new form of analysis paralysis. Organisations already possess shelves of process material assembled by programmes that never changed the work. Requiring comprehensive understanding before automation could protect incumbency, inflate discovery phases and allow every stakeholder to declare their exception indispensable.

That objection should change the method, not the principle.

Understanding need not mean documenting everything. It means reducing uncertainty in proportion to the authority and consequence of the automation. A tool that drafts an internal summary can safely proceed with less process certainty because a person remains responsible for the result. A system that initiates payment, changes a customer record or makes a consequential eligibility judgement requires much stronger evidence about decisions, information and controls.

The appropriate unit of discovery is not the entire end-to-end process. It is the decision boundary being automated. Teams can move quickly by taking a thin operational slice and testing it against real variation.

1. Select one outcome and define the authority the automation would receive.
2. Sample completed cases, including reversals, complaints and manual overrides.
3. Compare the documented path with observed work and transaction evidence.
4. Separate legitimate judgement from design debt and unmade decisions.

5. Automate only the portion whose decision rights and information limits are understood.
6. Use production feedback to widen the boundary deliberately, rather than allowing scope to expand through convenience.

In the composite refund operation, this discipline changed the ambition without killing it. The team did not attempt autonomous handling of all refunds. It automated data assembly and the calculation for six stable reason codes, covering 38 per cent of volume. Staff retained approval authority, but the median handling time for those cases fell from eighteen minutes to seven. More importantly, overrides were coded by cause. After eight weeks, two data defects accounted for 61 per cent of overrides. Correcting them allowed the automated boundary to expand on evidence rather than enthusiasm.

The smaller first release looked less impressive than the original demonstration. It created more durable capability because it made the organisation learn.

7. The real divide is not automated versus manual

Much of the current debate places manual work at one end of a line and autonomous work at the other. That framing encourages a race toward the far end. It also misses the deeper distinction.

The important divide is between work whose decisions are understood and work that continues through accumulated accommodation. Both manual and automated operations can exist on either side. A well-understood manual process may be a sound candidate for rapid automation. A highly digital process may remain structurally opaque because no one can explain why its exceptions, controls and ownership developed as they did.

Generative AI makes this distinction urgent because it is unusually capable at operating within ambiguity. Used carefully, that ability can help an organisation surface and examine variation. Used carelessly, it can let the organisation postpone comprehension while appearing to advance.

We should therefore judge an AI-enabled transformation by more than the fluency of its outputs or the number of steps it can execute. The harder evidence is organisational: clearer decisions, fewer disguised workarounds, better information boundaries and a functioning mechanism for learning from exceptions.

The organisations that cross the capability cliff will not be those with the least automation. They will be those that mistake automation for understanding. The more autonomous the technology becomes, the less tolerable that mistake will be.