

BEYOND FEASIBILITY

The Proof-of-Concept Graveyard: Why AI Rarely Survives the Journey to Production

On feasibility, funding, and the day-two problem — why the demo that dazzles is the least useful thing an experiment can prove

Giovanni Leonardi

April 2023



Contents

1. Executive Summary

3

2. The Demo That Dazzles

3

3. Why "It Works" Is the Least Useful Thing to Prove

4

4. The Structural Forces That Keep the Graveyard Full

5

4.1 Funding the theatre, not the path

5

4.2 The absent owner of day two

5

4.3 The antibodies of the operating model

6

4.4 The debt that surfaces at the boundary

6

5. Why This Wave Digs the Graveyard Faster

6

6. The Strongest Case for the Graveyard — and Its Limit

7

7. What Changes When the Practitioner Knows This

8

8. The Graveyard as a Mirror

9

The Proof-of-Concept Graveyard: Why AI Rarely Survives the Journey to Production

Transformation · No. 377

Originally written April 2023 | Re-edited and re-rendered for the WhereBeyond Edition, 2025

"The pilots that reach production are built by organisations that have already decided to be changed by them."

1. Executive Summary

Every wave of enterprise technology leaves a graveyard behind it, and the graveyard is always full of proofs of concept that worked. They worked in the demo, they worked in the notebook, they worked well enough to earn applause in a steering committee — and then they died on the threshold of production and were never spoken of again. We are living through such a wave now. The arrival of large language models that anyone can prompt has made the proof of concept cheaper, faster, and more seductive than at any point I can remember, and the predictable consequence is that organisations are generating pilots faster than they have ever generated them, and burying them faster too.

This essay argues that the proof-of-concept graveyard is not, and has never been, a technology problem. The technology is the part that works. What fails is everything the demo was designed not to test: the ownership of the thing once it is live, the data it will actually run on rather than the tidy sample it was shown, the governance it must satisfy, the process it must displace, and the accountability for its behaviour on the day it is wrong. A proof of concept proves feasibility. Feasibility was rarely the binding constraint. And so the same organisations that were unable to move machine-learning models out of the laboratory five years ago are now unable to move language-model pilots out of the sandbox, for reasons that have almost nothing to do with either technology and almost everything to do with how they are built to absorb — or reject — anything new.

The uncomfortable conclusion is that the pilot which reaches production is usually a symptom rather than a cause. It reaches production because the organisation around it had already done the harder work of deciding who would own it, what it would replace, and how it would be held to account. Where that work is absent, no model, however impressive in the demo, will survive the journey. The graveyard is a mirror. It shows us not the limits of the technology but the limits of the enterprise that commissioned it.

2. The Demo That Dazzles

I have watched the same meeting play out more times than I can count, and it has a rhythm to it. A small team, often assembled in a hurry, presents something built in a few weeks. It reads a stack of contracts and answers questions about them in plain language. It drafts a first-pass response to a customer complaint that is better than the one a tired human would write at four in the afternoon. The room is genuinely impressed, and the impression is not misplaced — the thing is impressive. A sponsor says the words that seal its fate: let's get this into production. Everyone nods. A slide is updated. And then, quietly, over the following two quarters, nothing happens, until the pilot is a line item in a review that no longer has a champion, and it is retired without ceremony.

The team is blamed, or the technology is blamed, or the vendor is blamed. All three explanations are wrong. What actually happened is that the demo answered a question nobody needed answered, and left every question that mattered untouched.

Consider a composite that is true to life. A pilot to triage inbound service requests reaches ninety-two per cent accuracy on a curated set of a few thousand historical tickets. On the strength of that number it is declared a success and slated for production. Then the real questions arrive. The historical tickets were clean; the live queue contains attachments, forwarded chains, three languages, and a category of request that did not exist when the sample was drawn. There is no monitoring, because a notebook does not need monitoring. There is no owner, because the team that built it was an innovation unit whose mandate ends at the demo. The service desk that would actually run it was not in the room, and when shown the tool, points out — correctly — that a wrong triage at scale creates more work than it removes, and that nobody has said who is accountable when that happens. The ninety-two per cent, so persuasive on the slide, turns out to have measured the one thing that was never in doubt.

A proof of concept proves that something can work. It almost never proves that we can run it, own it, govern it, and fold it into the way decisions are actually made. Those are the constraints that bind, and they are precisely the constraints a demo is designed to set aside.

3. Why "It Works" Is the Least Useful Thing to Prove

There is an old and useful distinction, borrowed from the language of product, between three questions any new capability must answer. Is it feasible — can it be built and made to work? Is it desirable — does anyone actually want the outcome it produces, enough to change their behaviour to get it? And is it viable — can the organisation sustain it, operate it, afford it, and live with its consequences? The proof of concept, as it is usually run, is an assault on the first question and a studied avoidance of the other two. Yet in an enterprise of any maturity, feasibility is the question least likely to be the reason something fails.

The model, in the end, is a small part of the system that surrounds it. The demonstrable, presentable piece — the part that reads the contract, drafts the reply, scores the ticket — is a slender fraction of what a production capability consists of. The rest is unglamorous and undemonstrated: the pipeline that feeds it live data in the shape it expects, the controls that stop it acting on data it should never have seen, the monitoring that notices

when its behaviour drifts, the fallback for when it is unavailable, the audit trail for when someone asks why it did what it did, the retraining or re-grounding when the world moves underneath it, and the human process redesigned around its existence. None of that appears in the demo. All of it appears in production. The graveyard is the gap between the two.

- The pilot proves the model reads a contract; production requires that every contract, in every format, from every source system, reaches the model reliably and lawfully.
- The pilot proves an answer is plausible; production requires a mechanism to catch the answer that is plausible and wrong, before it reaches a customer or a regulator.
- The pilot proves a person can operate the tool; production requires that the people whose work it changes will actually use it, and have a reason to.
- The pilot proves a capability exists; production requires that someone owns it on the day it breaks.

Each of these is a discipline in its own right, and none of them is technological in the interesting sense. They are organisational. This is why the teams that are best at building proofs of concept are so often the worst at shipping them: brilliance at the spike is no preparation at all for the long, dull, contested work of the last mile.

4. The Structural Forces That Keep the Graveyard Full

If this were merely a matter of teams forgetting to plan for production, it would have been solved long ago by a checklist. It has not been solved, which tells us the causes are structural — that the organisation is, in a real sense, built to produce and then abandon pilots. Several forces sustain it.

4.1 Funding the theatre, not the path

The first is a matter of money and what it is for. Innovation budgets and run budgets live in different places, answer to different people, and measure themselves differently. A proof of concept funded from an innovation line is measured by whether it produced a compelling demonstration and a good story. It was never funded to reach production, and reaching production would require it to cross an organisational and financial boundary that nobody has been made responsible for helping it cross. The pilot is complete, on its own terms, the moment the demo lands. That it does not continue is not a failure of the pilot; it is the design of the funding. We have paid for theatre and then expressed surprise that we got a performance rather than a building.

4.2 The absent owner of day two

The second is the day-two problem. A demonstration has an author; a production system needs an owner, and the two are rarely the same person or even the same function. Ownership on day two means being accountable for the capability when it drifts, when it is wrong, when it must be explained to a regulator or a board, when it needs the unglamorous maintenance that keeps any system alive. In most organisations, no such owner is named before the work begins, and by the time the question is asked — who runs this? — the innovation team has moved on, the operational team was never consulted, and

there is no home for the thing in the operating model. An orphan does not reach production. It waits in the corridor until it is quietly asked to leave.

4.3 The antibodies of the operating model

The third force is the most human and the least discussed. A capability that reaches production does not slot into empty space; it displaces something — a process, a set of tasks, sometimes a role, sometimes the tacit expertise that gave certain people their standing. Organisations have antibodies, and they are not irrational. The service desk that questions the triage tool is protecting a system it understands from a system it does not, whose failure modes are unknown and whose accountability is unassigned. Resistance of this kind is usually read as Luddism. It is more often a correct assessment of risk that the pilot never bothered to answer. When we treat the people whose work will change as obstacles rather than as the source of the requirements we forgot to gather, we guarantee the antibody response we then complain about.

4.4 The debt that surfaces at the boundary

The fourth is that data and governance debt is invisible until the production boundary forces it into the light. In the sandbox, the data is a clean extract, the privacy questions are deferred, the lineage is nobody's concern, and the model is asked only to perform. At the boundary, all of it comes due at once: the data is messier and more sensitive than the sample, the lawful basis for using it was never established, the third party the model calls out to would receive information it must not, and the controls that would make any of this acceptable do not exist. The debt was always there. The pilot succeeded precisely by not paying it. Production is where the invoice arrives.

"The pilot succeeds by not paying its debts. Production is where the invoice arrives."

5. Why This Wave Digs the Graveyard Faster

None of the above is new. The machine-learning wave of the previous decade filled a graveyard of its own, one stocked with models that reached impressive accuracy in a data scientist's notebook and never touched a customer. We invented an entire discipline — the operational plumbing of models, the monitoring of drift, the machinery of getting a trained thing to run reliably in the world — largely in response to how badly we had underestimated the last mile. The lesson was learned by the people who lived it and, characteristically, not institutionalised beyond them.

What the current wave has changed is the economics of entry, and it has changed them dramatically. Where the old proof of concept required scarce specialists, labelled data, and weeks of training, the new one requires an afternoon, an interface, and a plausible prompt. A capable model arrives pre-trained; the hardest technical step of the old world has been removed. This is genuinely transformative, and it is also why the graveyard is

filling faster than ever. When the cost of producing a proof of concept collapses, the number of proofs of concept explodes – and because none of the organisational costs of production have fallen at all, the ratio of pilots to production systems gets worse, not better. We have made the cheap part nearly free and left the expensive part exactly as expensive as it was.

The new wave also brings failure modes the old one did not, and they land squarely on the production side of the boundary. A model that generates fluent, confident, and occasionally fabricated output cannot be governed the way a classifier with a probability score could. The tool that answers in convincing prose is harder to bound than the one that returns a number, because its errors are more persuasive and its behaviour less predictable. There is the matter of information leaving the building: a pilot that sends text to an external service is, unless someone has thought hard about it, a channel for confidential and regulated data to go where it must not, and by early this year enough organisations had noticed to start restricting what their staff could paste into a public chatbot. And there is the question, still largely unanswered as I write, of who is accountable for what a generative system says – a question that production forces and the demo never raises.

So the paradox of this moment is real. The technology has never been more capable, the barrier to a proof of concept has never been lower, and for exactly those reasons the proportion that reaches production may well be lower than in any wave before it. Abundance at the front of the funnel does not help if the funnel has no outlet.

6. The Strongest Case for the Graveyard – and Its Limit

It would be dishonest to argue all of this without confronting the best defence of the graveyard, which is not a weak argument at all. Proofs of concept, the defence runs, are supposed to die. That is their entire purpose. A cheap, disposable experiment exists to answer a question and be thrown away; a high mortality rate among pilots is not a pathology but the signature of a healthy portfolio that is running many small bets and killing the ones that do not pay off. To lament that most pilots do not reach production is to misunderstand what a pilot is for. In a world where an experiment costs an afternoon, we should run more of them and expect to bury nearly all of them, and the organisation that ships every pilot it starts is not disciplined – it is failing to experiment at all.

This is correct, and it is important, and it does not rescue most real graveyards. The defence holds only under two conditions that are usually absent. The first is that the dying experiment leaves learning behind – that its death answers a question the organisation retains and reuses. The second is that the portfolio is actually designed to graduate some pilots – that there exists a path, funded and owned, along which the survivors travel to production. Where both conditions hold, a high kill rate is indeed health. Where they are absent, what looks like a portfolio of experiments is something else entirely: a sequence of orphans, each stalling for the same unaddressed reasons as the last, each teaching nothing because nobody captured what it taught, none graduating because no graduation path was ever built.

The distinction that matters, then, is between an experiment that concludes and an orphan that stalls. An experiment that concludes has a hypothesis, a result, and a decision

— including the decision to stop, made deliberately and recorded. An orphan that stalls has none of these; it does not conclude, it simply loses its champion and drifts out of memory. A hundred per cent mortality rate among orphans is not a healthy funnel. It is a broken one wearing the costume of experimentation. The defence of the graveyard is a defence of concluded experiments. It is no defence at all of orphans, and the graveyard is mostly full of orphans.

7. What Changes When the Practitioner Knows This

If the graveyard is organisational, the response must be organisational too, and it begins before a single line of the pilot is built. I do not offer a method here — the honest shape of this problem resists the checklist that has already failed to solve it — but there are shifts in disposition that separate the organisations whose pilots travel from those whose pilots are buried, and they are worth naming plainly.

1. Prove the hardest thing, not the easiest. A proof of concept should be aimed at whatever is most likely to kill the idea in production — the messy data, the governance constraint, the adoption question — and not at the capability that was never in doubt. If the demo is impressive, there is a fair chance it tested the wrong thing. The most useful pilot is often the one that looks least like a triumph, because it went looking for the reason to stop and either found it early and cheaply, or earned the right to continue.
2. Fund the path, not the spike. A pilot commissioned without a named, funded route to production is being commissioned as theatre, whatever the intent. The decision to explore and the decision to be willing to operate should be taken together, or the exploration should be understood, openly, as a concluded experiment rather than a candidate for production.
3. Name the day-two owner before day one. The question who will run this when it is live, and who is accountable when it is wrong? should be answerable before the work starts. If it cannot be answered, that is the finding — and it is a more valuable finding than any accuracy score.
4. Treat data and governance as the real proof of concept. The interesting risk almost never lives in the model. It lives in whether the data can lawfully and reliably reach the model in production, and whether the system's behaviour can be governed once it does. A pilot that proves those things, even with an unglamorous model, has proved something worth knowing. A pilot that proves a dazzling model on data it will never see again has proved almost nothing.
5. Measure adoption, not accuracy. The metric that predicts production is not how well the thing performs in a test but whether the people whose work it changes will actually use it, and why they would. That metric can only be gathered by involving those people from the beginning, as the source of the requirements rather than as the recipients of the result.

None of these is a technical intervention. Every one of them is a decision about ownership, funding, accountability, and the willingness to change the operating model rather than merely to demonstrate against it. Which is the whole point. The pilots that reach production are built by organisations that have already decided to be changed by them.

8. The Graveyard as a Mirror

Return, at the end, to that meeting — the demo, the applause, the sponsor's fatal enthusiasm. It is tempting to see the pilot that follows it into the graveyard as a story about technology that was not quite ready, or a team that did not quite deliver, or a vendor who over-promised. It is none of those. It is a story about an organisation that could produce a demonstration but had not decided to produce a change, and the pilot merely made that undecided state visible, at speed.

This is why the current wave is worth watching so closely, and why the temptation to measure it by the brilliance of its demonstrations is so misleading. The brilliance is real and it is also beside the point. When the cost of a proof of concept falls to nearly nothing, the demonstration stops being evidence of anything except that the technology works, which we already knew. The scarce thing — the thing that was always scarce — is the organisational will to own, govern, fund, and absorb a new capability into the way real decisions are made. That was the binding constraint in the machine-learning wave, it is the binding constraint now, and it will be the binding constraint in whatever wave comes next, because it was never a property of the technology at all.

The graveyard, then, is not a verdict on our models. It is a mirror held up to our organisations, and the reflection is the same in every era: we are fluent in building the thing that impresses and far less fluent in becoming the kind of enterprise that can live with it. The pilots that escape the graveyard are not the cleverest. They are the ones commissioned by organisations that had already done the harder, quieter, un-demonstrable work of deciding to change. Everything else is a demo, and demos, however dazzling, are where the graveyard begins.