PROGRAMME BOOK SERIES · No. 255

Perspective

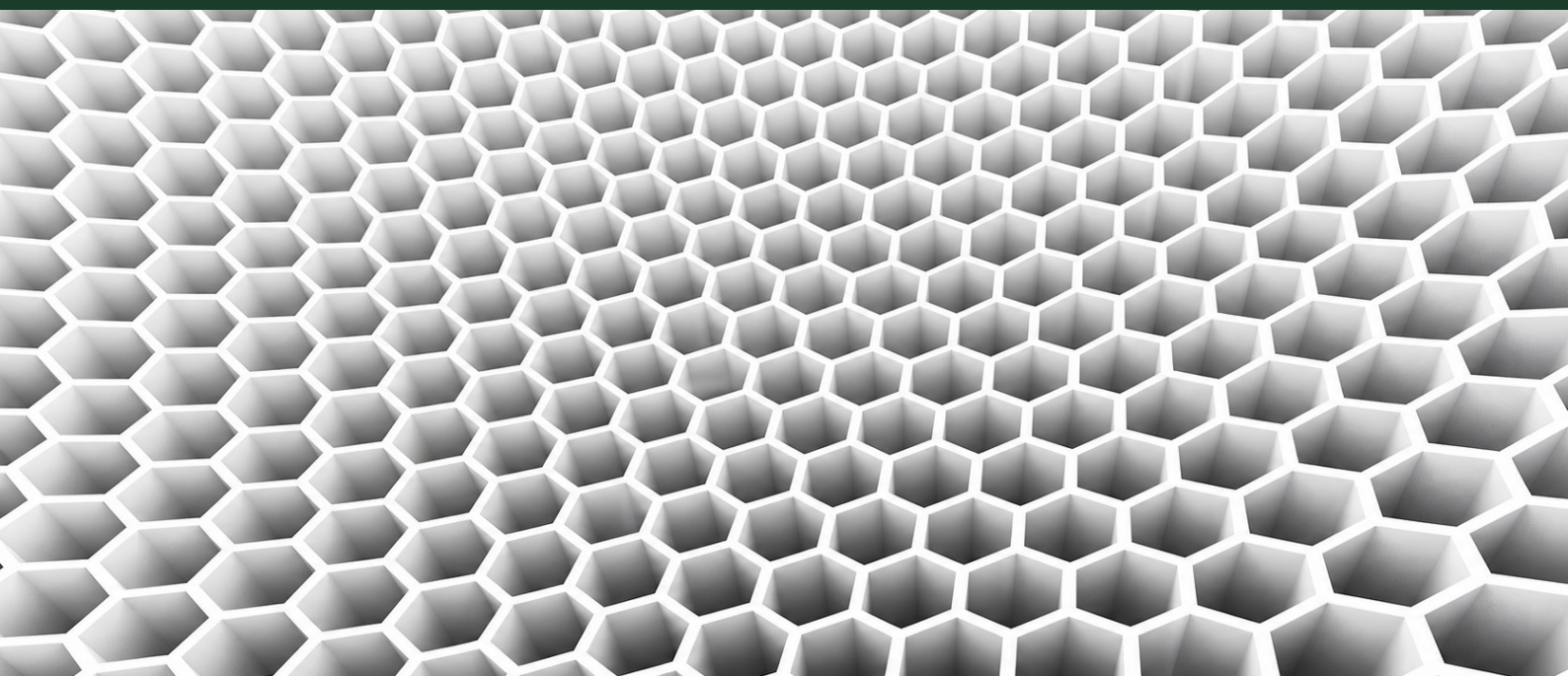
BEYOND ETHICAL ASSURANCE

Responsible AI Is Programme Scope — Not an Assurance Review at the End

The honest delivery lesson from early generative AI programmes: safeguards that are not planned, funded and tested will not survive contact with the deadline

Giovanni Leonardi

May 2023



Responsible AI Is Programme Scope — Not an Assurance Review at the End

Programme · No. 255
Originally written May 2023 | Re-edited and re-rendered for the WhereBeyond Edition,

"If responsible AI is not inside scope, the programme will experience it only as late risk, late cost and late opposition."

The Review Arrived After the Programme Had Already Decided

The artificial-intelligence pilot had reached its final demonstration. The team could show an 88 per cent success rate in routing customer enquiries, a projected reduction of nine minutes per case and a credible route to deployment. The sponsor expected approval to proceed.

Then the responsible-AI review began.

The reviewers asked which customer groups were misrouted most often, how a customer could challenge the outcome, which version of the prompt and model had produced each response, what data had been retained, and who would suspend the service if behaviour changed. The programme had partial answers, scattered across technical notes and workshop minutes. None had been treated as a delivery requirement.

Four weeks later, the launch was delayed. The responsible-AI team was blamed for arriving late. In truth, responsibility had arrived late because the programme had placed it there.

This is the practitioner's uncomfortable account of responsible AI in 2023: organisations say it is a principle, create a framework, and then manage it as an assurance activity outside the programme's real scope. Delivery is funded to prove the use case. Responsibility is invited to comment on what has already been built.

That sequence guarantees conflict.

If responsible AI is not inside scope, the programme will experience it only as late risk, late cost and late opposition.

Principles Do Not Compete with Milestones

Responsible-AI frameworks commonly express sensible principles: fairness, transparency, privacy, safety, accountability and human oversight. The difficulty begins when those principles enter a delivery plan.

A principle has no natural owner. It does not estimate itself, create an interface, define an exception process or reserve time for testing. Unless translated into work, it remains morally important and operationally optional.

The programme feels the pressure asymmetrically. The demonstration date is visible. Model performance is measurable. Supplier cost is tracked. The ability of a customer to contest an automated recommendation is less visible until somebody needs it. Bias monitoring appears to slow development; the absence of bias monitoring does not slow the plan at all.

This is why good intentions lose.

It is not normally because delivery teams oppose responsible practice. It is because governance asks them to optimise against funded scope, and responsibility has been expressed as expectation rather than scope. Under time pressure, explicit deliverables defeat implicit obligations.

The result is a familiar progression:

- a framework is approved at enterprise level
- a programme interprets it locally, often without specialist capacity
- controls are documented in a checklist
- unresolved questions are accepted for the pilot
- the pilot creates momentum and executive expectation
- deployment exposes operational, legal or reputational concerns
- assurance becomes the apparent blocker

The failure began before the first model was configured. The business case defined value without pricing the conditions under which that value could be pursued responsibly.

What Scope Looks Like in Practice

Treating responsible AI as scope does not mean adding a general “ethics workstream”. That often creates another boundary: specialists write principles while engineers and operational teams make the consequential choices.

The work must be attached to the use case.

For a customer-triage system, responsible scope might include:

- a documented purpose and a list of prohibited uses
- performance measures split by relevant customer and case groups
- an explanation suitable for the employee acting on the recommendation
- a route for customers and staff to challenge or correct an outcome
- records connecting data, model version, prompt, output and human action
- limits on what information may enter the service
- monitoring thresholds that trigger review, restriction or suspension
- a named operational owner with authority after the project closes

Each item changes delivery. Group-level testing may require better labels. A challenge route requires process design and staffing. Traceability affects architecture. Suspension thresholds require operational judgement. Data restrictions shape user guidance and technical controls.

That is the point. Responsibility is not commentary on the solution. It is part of the solution.

The Composite Failure Hidden by an Average

Consider a composite programme using a generative model to draft responses for an internal advisory service. The pilot reviews 6,000 historical queries and reports that 84 per cent of drafts need only minor editing. On average, handling time falls from eighteen minutes to eleven.

The aggregate looks strong.

A closer sample shows that performance falls sharply for rare policy exceptions. These cases represent only 7 per cent of volume but nearly half of material corrections. Experienced advisers recognise them from context; the model produces fluent answers that apply the standard rule.

The programme has three practical choices: exclude exceptions from scope, create a reliable method for identifying them, or require enhanced human review. Each choice reduces the headline benefit. None is a reason to abandon the use case.

But because exception handling was not in the original scope, the choice arrives after the benefit has been socialised. Responsible delivery now looks like value destruction. The

board asks why the issue was not raised earlier. The team answers that it needed a working model before it could know.

That answer is partly fair. Some risks emerge only through use. The mistake was not failing to predict the 7 per cent. It was failing to reserve scope, time and authority for what would be learned.

Responsible AI cannot be a promise that nothing unexpected will happen. It must be the programme's capacity to detect, decide and respond when it does.

The Serious Objection: Do Not Govern the Experiment to Death

There is a strong opposing case. Generative AI is moving quickly. Its limitations are not fully understood, and the only credible way to learn is through controlled experimentation. If programmes must specify every safeguard before they begin, governance will freeze immature assumptions, favour large suppliers and prevent useful discovery. Excessive documentation can create the appearance of control without improving outcomes.

That objection is correct about rigid front-loaded assurance.

Responsible AI should not require a pilot to meet the same controls as a live service affecting thousands of people. Nor should every use case carry identical governance. A tool helping an employee brainstorm headings does not create the same consequence as one influencing access to a service.

But proportionate governance is not absent governance. Early scope should define the boundary of the experiment, the data it may use, the people exposed, the evidence to collect and the decision that follows. As consequence increases, so should the burden of proof.

> *The answer to uncertainty is not to defer responsibility; it is to make learning itself governed programme work.*

Put Responsibility on the Critical Path

Programme leaders should make four changes.

1. Write the responsible operating conditions into the business case. State who may be affected, what harm matters, what human accountability remains and what continuing controls will cost.
2. Turn principles into acceptance criteria. A principle becomes real when the programme can test it, assign it and refuse deployment if it is not met.
3. Fund discovery and rework. Reserve capacity for data investigation, adversarial testing, exception design, user feedback and changes prompted by evidence.
4. Create a live-service owner before launch. Model behaviour, user practice and source data will change. Someone must own monitoring and have authority to restrict or stop the service.

These actions move responsible AI from the edge of governance to the centre of delivery. They also improve programme honesty. Benefits can be presented alongside the cost of oversight. Pilots can succeed by disproving an unsafe assumption, not only by securing deployment. Assurance can challenge while choices are still reversible.

Responsibility Is a Delivery Capability

The early response to generative AI has produced an abundance of principles and urgency. Both are understandable. Neither is enough.

Frameworks matter because they give organisations a language for concerns that otherwise remain diffuse. But frameworks do not deliver responsibility. Programmes do—through requirements, architecture, testing, operating processes, funding and accountable decisions.

The honest lesson is that responsible AI will always feel expensive when the programme pretends it is free. It will always feel late when it is scheduled after the technology. It will always feel obstructive when benefits are promised before safeguards are priced.

We should stop asking assurance to rescue programmes from choices already embedded in their scope.

Responsible AI belongs wherever the programme defines value, designs the service, accepts risk and decides to launch. Put it there from the beginning, and it becomes a discipline of delivery. Leave it outside, and it will return at the end as the cost of everything the programme chose not to see.