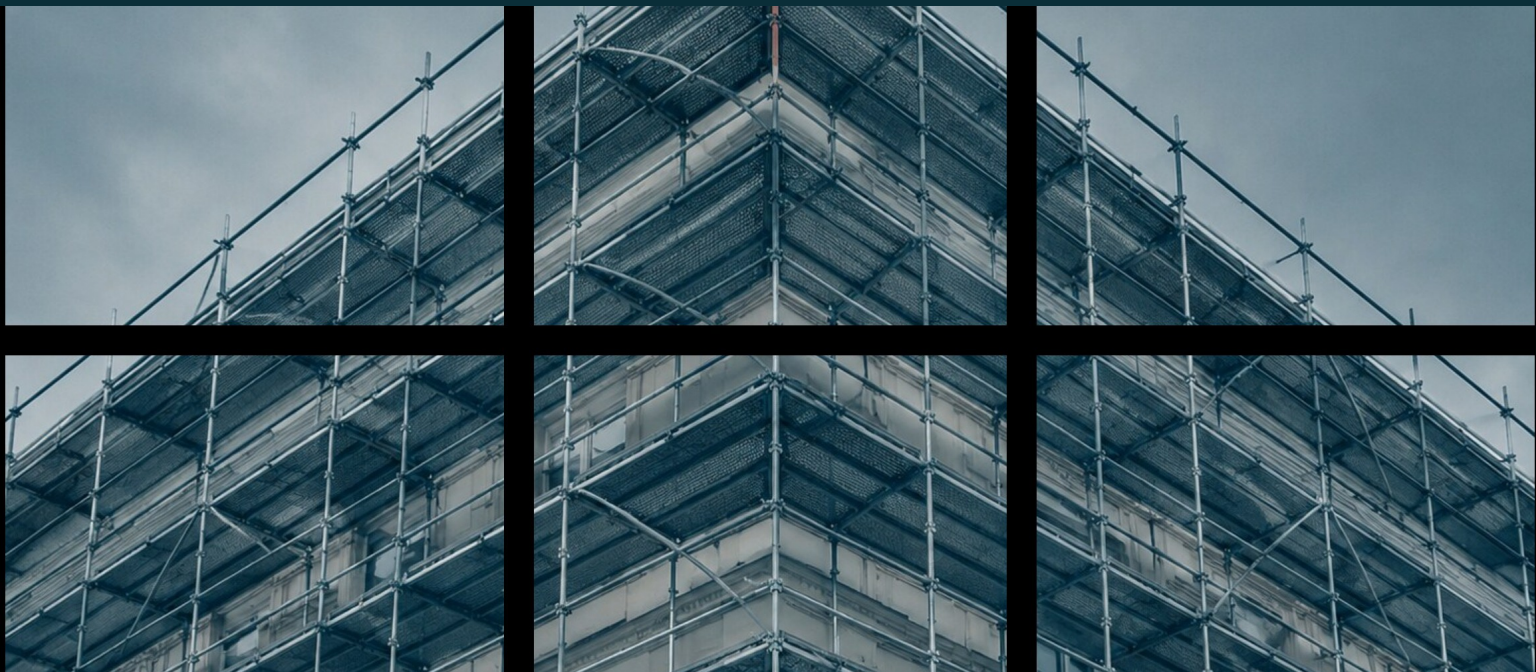


BEYOND THE DOCUMENT

What Agile Took From the Business Analyst — and What It Exposed

On the profession that lost its central artefact in 2011, and discovered what its value had actually been.



Giovanni Leonardi

August 2011

What Agile Took From the Business Analyst – and What It Exposed

Transformation · No. 165

Originally written August 2011 | Re-edited and re-rendered for the WhereBeyond Edition, 2025

“The document was never the job. It was where the job used to live.”

1. The stand-up you have no line in

There is a particular silence that a business analyst learns to recognise this year. It happens about four minutes into a stand-up, when the team has gone round the circle – the developers, the tester, the product owner reading from a backlog she now owns – and the analyst realises there was no natural place for them to speak. Not because they had nothing to say. Because the artefact that used to be theirs, the one that gave them a reason to be in every conversation, has been quietly dismantled and handed out in pieces to other people.

For a decade the business analyst's place in a programme was guaranteed by a document. The requirements specification was long, it was signed, and it was the hinge on which the whole delivery turned. Nothing was built until it existed; everything built was measured against it. Owning that document meant owning the traffic between the business and the build, and no one questioned whether the role was necessary because the artefact so obviously was.

That artefact is now being taken apart in front of us. The requirements are a backlog, and the product owner holds it. The detail is in user stories written three at a time, a week before they are built. The acceptance criteria that used to live in section 7 of a signed specification are now half a dozen bullet points a developer and a tester thrash out at a whiteboard. The "three amigos" conversation has a business voice in it, but that voice is increasingly the product owner's, not the analyst's. Strip away the document and a quiet, frightening question surfaces for a great many capable people: if the specification was the job, and the specification is gone, what exactly am I now?

2. Two things the document was doing at once

To answer that honestly you have to see what the requirements document actually did, because it was doing two quite different jobs and hiding the difference between them.

The first job was analysis: understanding what the business was really trying to achieve, which was almost never what it first asked for; finding the process underneath the request; noticing the exception that would break in production; deciding what mattered and what was noise. This is thinking, and it is hard, and it is the part that took years to get good at.

The second job was specification-as-contract: writing that understanding down in a form complete and unambiguous enough that it could be signed, handed across an organisational boundary to a delivery team who might be in another building or another

country, and later used to adjudicate whether what was built was what was agreed.

The document fused these two so tightly that, from the outside, they looked like one skill. And here is the uncomfortable truth the profession has not quite said aloud: the fusion meant that an analyst who was excellent at the second job and merely adequate at the first could pass for a good analyst for an entire career. If your pages were thorough, your traceability matrix complete, your sign-off clean, the quality of the thinking underneath was very hard for anyone to inspect. The artefact was a place to hide.

“Agile did not make the analyst redundant. It removed the document behind which two very different analysts had been indistinguishable.”

3. The honest case that the role is finished

It is worth stating the opposing view at its strongest, because the weak version — “agile is just cowboys who hate documentation” — is a caricature that lets analysts off the hook.

The strong version goes like this. The requirements document was not a neutral good; it was a symptom of distance. You needed a heavy, signed specification precisely because the people who understood the business and the people who wrote the code were kept apart — different teams, different buildings, a contract between them. Agile closes that distance on purpose. It puts a product owner with real business authority inside the team, permanently. It replaces the up-front contract with a continuous conversation and a working increment every couple of weeks. And once the business voice is in the room every day, the intermediary who used to carry meaning between the two sides has no gap left to stand in. The product owner decides what matters; the developers and tester work out the detail directly with her; the increment shows everyone what was actually meant. On this reading the analyst was scaffolding for a building that has now been built, and taking the scaffolding down is not a loss but the point.

This argument deserves respect because a good deal of it is correct. The distance really was the disease. The document really was a symptom. And a product owner who is genuinely embedded and genuinely empowered really does absorb a large part of what a weaker analyst used to do.

4. What the argument gets wrong

But it mistakes the artefact for the capability, in exactly the way the profession itself did.

Closing the distance between business and build does not remove the need for analysis. It disperses it and makes it continuous. The questions the analyst used to answer once, up front, over six weeks, now arrive in fragments, every day, at speed. Which of these forty backlog items actually delivers the outcome the sponsor is funding, and which are there because someone loud asked? When this epic is broken into stories, where is the seam that will quietly drop a regulatory obligation on the floor? This story looks trivial to the developer — does he know it touches the one process where getting it wrong is a

reportable breach? The product owner has an opinion on priority; does she know the second-order effect on the downstream team that no one in this room represents?

That is analysis, and the pace of agile does not abolish it. It raises the tax on getting it wrong, because there is no eleven-week sign-off gate downstream to catch the error before it ships. The gate is gone. The increment goes live. Whatever thinking did not happen becomes a defect in production, or worse, a defect that passes every test because it does precisely what the poorly-understood story asked.

Consider a case I have watched closely. A programme's requirements specification ran to something over three hundred pages and took eleven weeks to move from draft to signed. When it turned to iterative delivery, the instinct was to declare the analyst role finished — the document was gone, so surely the documenter was too. What actually happened is that the analyst who understood the business rather than merely the format became more valuable, not less. She stopped writing three hundred pages nobody read to the end and started doing something narrower and harder: sitting in backlog refinement and being the person who could say, in one sentence, why a story mattered, what broke if it shipped wrong, and which invisible exception it was quietly stepping on. The three-hundred-page artefact was pure loss. The judgement that had gone into it was the entire value, and it transferred cleanly to a world with no document in it at all. The analysts beside her whose value had genuinely been the three hundred pages — the completeness, the formatting, the traceability — did not transfer, because there was nothing underneath to move.

5. The identity worth having

So the crisis is real, but it is not the crisis it appears to be. The role is not dying. What is dying is a particular basis for the role — an identity built on ownership of an artefact rather than on a capability.

That is the genuinely painful part, and it is why the anxiety in the profession this year is not misplaced hysteria. An identity anchored to a document is fragile, because a document can be taken away in a single change of method, and this one just was. An identity anchored to a capability — the ability to understand what a business actually needs, to find the exception that matters, to turn intent into something a team can build without losing the meaning on the way — is portable, because it survives any change in the container that capability happens to be poured into. It survived the move from use cases to user stories. It will survive whatever replaces user stories.

The advice that follows is unglamorous but exact. Stop defending the document; it is not coming back, and defending it marks you as someone whose value was the format. Get into the room where the thinking now happens — refinement, the three amigos, the messy conversation where a vague story becomes a buildable one — and make yourself the person there who reliably sees what others miss. The title on the badge is not the point. Whether it still reads "business analyst" in five years is genuinely uncertain. Whether organisations building software at speed will still need people who can do the analysis is not uncertain at all. They will need it more, and more urgently, precisely because they have taken away the long slow gate that used to hide the cost of doing it badly.

The document was never the job. It was where the job used to live. The job is coming with us.