

BEYOND THE COMPLIANCE PROJECT

When the Regulator Became the Product Owner

Why the year's defining data regulation is not a compliance project but the most demanding stakeholder your delivery teams have ever had

Giovanni Leonardi

February 2018



When the Regulator Became the Product Owner

Transformation · No. 214

Originally written February 2018 | Re-edited and re-rendered for the WhereBeyond Edition, 2025

“A forty-item control register is not fourteen weeks of progress; it is fourteen weeks of describing the problem.”

1. The programme that was green

Every reporting line was green. The data-protection workstream had a plan, a RACI, a risk register with forty-odd entries, and a milestone burn-down that told a reassuring story. Fourteen weeks out from the new regulation taking effect, the steering committee could look at the pack and feel that the organisation was on top of it.

Two floors down, the teams actually building the customer platform had not changed a line of code. Consent was still a single checkbox captured at sign-up and never revisited. The customer record still flowed, unremarked, into a dozen downstream systems that no one had mapped. The workstream was green because it was measuring itself — its documents, its meetings, its register — and not the systems the regulation was actually about.

This is the gap worth writing about, because it is the most instructive thing about the run-up to the new data regime. The organisations that struggled were not the ones that misread the law. They were the ones that treated a change to how their products must behave as a compliance project running alongside delivery, rather than as what it had quietly become: a new and extremely demanding product owner, sitting in the backlog, writing acceptance criteria nobody could descope.

2. A stakeholder you cannot descope

Consider what the regulation actually does, stripped of the legal register. It says: capture only the data you need. Be able to explain, per field, why you hold it. Let people see it, correct it, take it elsewhere, and have it erased. Ask for consent in a way that is specific and revocable, and prove you asked. Notify within seventy-two hours when something goes wrong. Design for all of this from the start, not as a bolt-on.

Read that as a lawyer and it is a set of obligations. Read it as a delivery lead and it is something more familiar and more uncomfortable: a backlog of user stories, with

acceptance criteria, written by someone outside the building, with a hard release date and no appetite for negotiation. "As a data subject, I can withdraw consent for marketing without withdrawing consent for service" is a product requirement. So is "the system can produce everything it holds about one person within a month." These are not risks to be logged. They are features to be built.

The regulator does not behave like a compliance department. It behaves like a product owner — it sets priorities, writes acceptance criteria, and defines what done means — and unlike your other stakeholders, it cannot be talked down at the trade-off meeting.

Once you see the regulation this way, the parallel-programme model looks obviously wrong. You would never manage your most important product owner by standing up a separate team to write a document about what they want, storing it in a register, and reporting green on the register while the product ignored them. Yet that is precisely how much of the market approached the deadline: a compliance function producing artefacts, and delivery teams who would meet the requirement only if, and when, someone translated it into their language and lifted it above the line in their sprint.

3. Why the seam goes unowned

The failure is structural, not a lack of effort. It comes from the fact that compliance and delivery sit in different worlds with different currencies.

- Compliance is measured on assurance — policies written, controls documented, sign-offs obtained. Its instinct is to produce evidence that the organisation intends to comply.
- Delivery is measured on flow — features shipped, throughput, cycle time. Its instinct is to protect the sprint from anything that arrives without a clear owner and a clear priority.
- Between them is the seam: the work of turning an obligation into a built, tested, released behaviour. That seam belongs to no one by default, and work that belongs to no one does not get done — it gets reported.

The tell is always the same: a register that grows more detailed as the deadline approaches while the systems it describes stay still. The detail is real work — genuinely careful analysis of what the law requires — but it accumulates on the wrong side of the seam. It documents the gap with increasing precision instead of closing it. A forty-item control register is not fourteen weeks of progress; it is fourteen weeks of describing the problem.

The deeper reason the seam is dangerous is that regulatory requirements are exactly the kind that look cheap from a distance and turn out to be expensive up close. "Support consent withdrawal" is one line in a register. In a real system it means finding every place consent was assumed, every downstream flow that inherited data on that basis, and every batch job that would happily keep processing after the customer said stop. You do not discover that cost until someone on the inside of the code goes looking — which is another way of saying you do not discover it from the register at all.

4. Regulatory intent as product requirement

The organisations that coped did one thing differently, and it was not working harder on compliance. They dissolved the parallel programme and moved the requirements into the product backlog, owned by the same people who owned everything else the product did.

Concretely, that meant three changes.

1. The obligations were rewritten as user stories in the teams' own backlog, prioritised against – and often above – feature work, with the same acceptance criteria and definition of done as anything else. Not a compliance plan referenced by delivery, but stories in delivery.
2. The data-protection specialist stopped being a gatekeeper at the end and became an advisor at the start – sitting with the teams, helping write the criteria, answering "does this satisfy the intent" in the room rather than in a review three weeks later. The expertise moved to where the decisions were made.
3. The work started from the data, not the document. Before writing a line, teams mapped where personal data actually lived and flowed – the one activity the register-driven approach reliably skips.

That last point is where the real work surfaced. On one platform I saw closely, the exercise of mapping data flows to support erasure and portability revealed that customer records were being copied into fourteen systems, four of which no current team could fully account for, and that a meaningful share of the stored data had no identifiable lawful basis at all – it was there because a form once asked for it and nothing had ever removed it. None of that was visible from the compliance register. All of it was visible the moment someone treated "the customer can be erased" as a feature that had to actually work.

The compliance-project view	The product-requirement view
Obligation logged in a register	User story in the team's backlog
Reported as a RAG status	Demonstrated as working software
Data-protection lead signs off at the end	Data-protection lead advises from the start
Measures intent to comply	Measures behaviour of the system
Discovers cost at audit	Discovers cost at design

The difference in outcome was not subtle. The register-driven teams arrived at the deadline with thorough documentation of everything they had not yet built. The backlog-driven teams arrived with fewer documents and more working behaviour, because they had spent the same weeks closing the seam instead of describing it.

5. The objection worth taking seriously

There is a serious argument against everything above, and it deserves better than a straw man.

The objection runs like this: a regulator is not a product owner and should never be treated as one. A real product owner obsesses over user value and makes trade-offs to maximise it; a regulator obsesses over risk and makes no trade-offs at all. Hand your backlog to that mindset and compliance will crowd out the customer, every story will acquire a legal veto, and delivery will slow to the pace of the most cautious reviewer in the building. Better, the argument goes, to keep compliance as a boundary — a set of constraints delivery must respect — and let product owners get on with creating value inside it.

This deserves to be taken seriously, because its failure mode is real. A literal version of "the regulator is your product owner" — where legal must approve every story and nothing ships without a compliance sign-off — genuinely does grind delivery to a halt. The point is not to hand the backlog over.

The point is that the choice the objection offers — regulation as constraint versus customer value — is a false one. Look at what the regulation actually asks for: hold less of my data, tell me why you have it, let me take it back, do not keep processing it after I have said no. Those are not the regulator's preferences imposed on the customer. They are the customer's own interests, written down by someone with the standing to insist. Treated as constraints, they arrive as friction. Treated as requirements, they are among the few items in the backlog you can be certain a user would actually want. Minimisation is not only lawful; it is less data to secure, migrate, and pay for. Purpose limitation is not only compliant; it is a cleaner data model. The regulation, read generously, is a proxy for product hygiene the organisation should have wanted anyway.

"The regulator is not competing with your customer for space in the backlog. On the questions the regulation covers, it is the only party in the room reliably speaking for them."

So the synthesis is neither "obey the regulator" nor "keep the regulator at the boundary." It is to internalise the regulatory intent as first-class product requirements, owned by the product, with the specialist as advisor rather than gate. That keeps the value orientation the objection rightly defends, and it closes the seam the boundary model leaves open.

6. The capability, not the deadline

It would be a mistake to file all this under one regulation and one date. The specific deadline matters enormously right now — it is fourteen weeks away and the work is real — but the deeper shift is that regulation has become a permanent, active stakeholder in how software behaves, not a periodic hurdle to clear. Data is only the first arena in which an outside party has started writing acceptance criteria for our products. With open

banking now live and the question of what leaving the EU will mean for data flows still unanswered, it plainly will not be the last.

Which means the capability worth building is not compliance with this regime. It is the muscle of translating regulatory intent into product requirements quickly and without drama — the practised ability to take an obligation, read it as the behaviour a user would want, and place it in the backlog next to everything else. The organisations that build that muscle now, under the pressure of a hard deadline, will find the next regime a matter of routine. The ones that stand up another parallel programme will spend the coming years reporting green on registers while their products, two floors down, stay exactly as they were.

The regulator has taken a seat at the table. The only real question is whether we hand it a document to file, or a backlog it can actually shape.