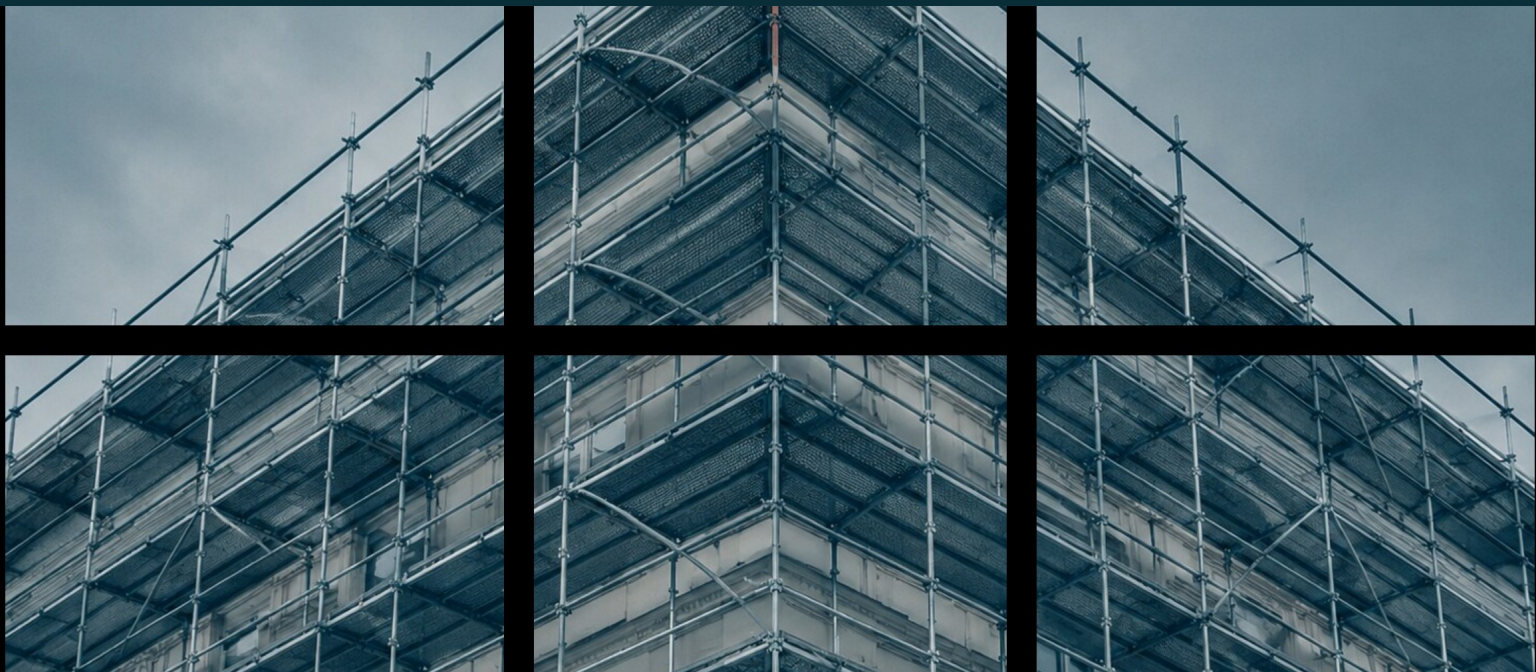


BEYOND DESIGN VERSUS RETROFIT

Built to Change, Not Built to Comply: The False Choice Between Compliance by Design and Retrofit

Why the real determinant of compliance cost is not when you build the control in, but whether you can change it later



Giovanni Leonardi

January 2007

Built to Change, Not Built to Comply: The False Choice Between Compliance by Design and Retrofit

Transformation · No. 121

Originally written January 2007 | Re-edited and re-rendered for the WhereBeyond Edition, 2025

“You cannot design in a requirement you do not yet have.”

1. The retrofit nobody budgeted for

Every practitioner who lived through the last few years of control remediation carries the same scar. A control obligation lands, and a system that was built years earlier — cleanly, competently, to the requirements of its day — turns out to have the offending logic threaded through it like a grain through wood. The reporting rule, the retention period, the segregation of a duty: none of it was wrong when it was built. It simply cannot now be changed without touching a hundred places at once. What should have been a configuration change becomes a re-engineering project, and a line that read “minor regulatory update” on the portfolio turns into the item that quietly consumes a year.

Out of this shared pain the industry has settled on a comfortable moral. Compliance by design — anticipating the obligation and building it in from the first line of code — is virtuous and cheap. Compliance by retrofit — bolting the control on afterwards — is negligent and expensive. The textbooks draw the two columns, put ticks against the first and crosses against the second, and leave it there. It is a tidy story, and like most tidy stories about a messy subject it is wrong in a way that matters.

*“The design-versus-retrofit debate is an argument about when.
The thing that actually governs the cost is whether you can
change it — and that is an argument about architecture.”*

I want to argue a single point in this piece: that the choice the textbooks pose is largely a false one, and that chasing “design” as a virtue in its own right leads organisations to the wrong investment. The variable that decides whether a control costs a fortnight or a year is not the moment you install it. It is whether the thing you built can absorb change at all.

2. Why “by design” is mostly unavailable

The case for compliance by design rests on a quiet assumption: that you knew the requirement when you were building. Sometimes you did. The obligations that have been stable for a decade — the core of prudential reporting, the settled parts of client money protection — can and should be designed in, because they were knowable and they held still.

But look honestly at the obligations that actually hurt, and most of them fail that test. They arrived after the system did. The system that is now non-compliant with a markets directive coming into force this year was specified when that directive was a consultation paper with three plausible shapes. The general-ledger platform that groaned under the weight of the internal-control attestation work of the last two years was commissioned before anyone in the building had heard the phrase. You cannot design in a requirement you do not yet have. To praise "compliance by design" as though the requirement were always sitting there waiting to be anticipated is to praise a foresight almost nobody possesses.

This is the first thing the textbooks leave out. The purest form of design-in compliance is available only for the obligations that were already stable — which are precisely the obligations that were never going to hurt you. For everything else, some form of after-the-fact accommodation is not a failure of discipline. It is the normal condition of building systems that outlive the rules they were built under.

3. Why retrofit is sometimes the right answer

The second omission is more uncomfortable, because it cuts against the moral entirely. There are obligations for which retrofit is not the regrettable outcome but the correct decision.

Consider a rule that is genuinely uncertain — under consultation, contested, likely to change shape before it settles, and possibly to be watered down or withdrawn. Designing your systems around the current draft is not prudence; it is committing capital to a specification that has not stabilised. The organisation that waits, keeps the obligation on a watch-list, and retrofits once the rule is final has not been negligent. It has declined to build twice. I have watched a firm spend heavily to embed the early draft of a reporting standard, only to rebuild the lot when the final text moved the goalposts — while a slower competitor, who had done nothing but keep the item under review, implemented the settled version once and at half the cost. Eagerness to design in the uncertain is its own expensive mistake, and the textbooks, fixated on the virtue of earliness, never mention it.

So the honest picture is not a good column and a bad column. It is a judgement, taken obligation by obligation, about how stable the requirement is and how expensive it will be to change the system later. And that second half of the judgement — how expensive to change later — is the one thing an organisation can actually do something about in advance.

4. The variable that actually matters

Here is the reframing the practitioner learns and the textbook misses. The expensive retrofit and the cheap retrofit are not distinguished by the timing of the work. They are distinguished by the architecture of the thing being changed.

When the offending logic is scattered — the retention period hard-coded in forty programs, the reporting rule reimplemented in every interface, the tax treatment baked into the core rather than expressed once and referenced — then any change, early or late, is expensive, because change means finding and altering every copy. When the same logic is isolated — the rule expressed in one place, the calculation held in configuration rather than code, the boundary between the stable core and the volatile regulatory layer drawn

deliberately — then even a late, unanticipated obligation is absorbed cheaply, because there is one place to change and the change is contained.

The goal is not to build systems that are compliant with today's rules. It is to build systems that are cheap to make compliant with tomorrow's — which is a property of structure, not of timing.

This is why the design-versus-retrofit framing sends investment to the wrong place. It tells you to spend your effort predicting the next obligation and building it in — a bet on foresight most firms will lose. The better counsel is to spend that effort on changeability: isolating the parts of the system most likely to be touched by regulation, pushing volatile rules out of hard code and into configuration, drawing clean seams around the regulatory surface so that when the unanticipated rule arrives — and it will — the retrofit is a fortnight's work in one module rather than a year's excavation across the estate. Some of that discipline travels under the banner of service orientation now much discussed; but the principle is older than any current label and does not depend on it.

The difference is not marginal. The same regulatory change — an alteration to how a class of transactions must be recorded and reported — will, in a well-partitioned system, be a configuration update tested in weeks. In a system where that logic was smeared across the estate, it is a multi-quarter programme with its own steering group. Same rule, same firm size, same competence of the people doing the work. The entire difference is in a structural decision taken, or not taken, years before the rule existed.

5. The strongest objection — and the answer

The serious counter-argument deserves stating at its strongest, not as a straw man. It runs: designing for changeability is not free. Building the extra layer of abstraction, the configuration engine, the clean seam, costs time and money up front, and adds complexity that has to be maintained forever. Much of that generality is never exercised, because most anticipated change never arrives. You are, the objection goes, paying a certain cost now to insure against an uncertain cost later — and disciplined engineering has always been suspicious of speculative generality built for a future that may not come.

The objection is right about the cost and wrong about the conclusion. It is true that you cannot make everything changeable; a system built as pure configuration everywhere is an unmaintainable horror that trades one paralysis for another. But that is an argument for placing the flexibility, not for abandoning it. The practitioner's craft is knowing where volatility lives — which surfaces regulation actually touches, again and again, across every regime — and buying changeability precisely there while building the stable core plainly and cheaply. Reporting formats change; retention rules change; classification boundaries change; the definition of a reportable event changes. You do not need to guess which change is coming to know where it will land. That is a far cheaper and far more reliable bet than trying to design in the specific rule, and it is available to every organisation whether or not it can see the next obligation coming.

6. What this asks of us

The shift the argument demands is less technical than it sounds; it is really a shift in what we treat as a virtue. We have been taught to admire the system that is compliant on the day it ships. We should instead admire the system that is indifferent to which rule comes next — that meets today's obligations plainly and can meet tomorrow's without being torn open. The first is a snapshot; the second is a capability.

For those of us who commission and govern these programmes, the practical consequence is a change in the questions we ask. Not only "does this design meet the current requirement?" — a low bar that any competent build will clear — but "when the requirement changes, as it will, how much of this will we have to touch?" That second question rarely appears on a design review. It is the one that decides, years later, whether a regulatory update is a line item or a crisis. The debate about design versus retrofit was always an argument about the wrong thing. The obligation we cannot yet name is already on its way; the only question that matters is whether we have built something ready to receive it.