

BEYOND PROCESS COMPLIANCE

Process Compliance Became a Substitute for Delivery

Why perfect governance evidence can conceal an unworkable outcome



Giovanni Leonardi

July 2007

Process Compliance Became a Substitute for Delivery

Programme · No. 87

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

“The programme is compliant at every checkpoint and wrong in the space between them.”

1. The Process Was Followed. The Outcome Was Lost.

By 2007, many large programmes had become exceptionally good at demonstrating that work had been governed. They could produce stage-gate packs, signed design authorities, complete risk registers, controlled change requests and immaculate audit trails. What they could not always produce was a working outcome at the point the organisation needed it.

This was not simply bureaucracy imposed by people who did not understand delivery. Process orthodoxy grew from legitimate experience. Major programmes had failed through weak controls, undocumented decisions, uncontrolled scope and heroic dependence on a few individuals. Regulators expected evidence. Boards wanted comparability. Suppliers needed contractual clarity. Programme leaders therefore built systems that made decisions visible and repeatable.

The problem began when evidence of control became evidence of progress. A completed procedure answered the question, "Did we follow the agreed method?" It did not answer the harder question, "Is the programme becoming more likely to deliver the intended result?"

That distinction sounds obvious. In practice, it is easily erased.

2. Why Compliance Became So Attractive

A process measure is reassuring because it is countable. A steering committee can see that 94 per cent of required documents have been approved, all workstreams have submitted weekly reports and every high-rated risk has an owner. These facts can be consolidated across a programme and compared from one month to the next.

Delivery evidence is more awkward. A new operating process may work in one region but fail under the volume of another. Users may complete training yet still revert to spreadsheets. Interfaces may pass formal testing but produce delays at month-end. Benefits may depend on managers changing decisions that no technical milestone can compel.

Process metrics therefore travel upwards more easily than operational truth. They compress complexity into a form that fits the governance calendar. The more distant a decision-maker is from the work, the more useful that compression appears.

There is also a contractual reason. In multi-party programmes, procedure defines who must do what, when acceptance occurs and where liability rests. A supplier can prove that

a deliverable met the documented entry and exit criteria. A client can prove that approval was withheld for stated reasons. Both sides need this discipline.

The strongest case for rigorous process is therefore substantial: without it, a complex programme can become arbitrary, untraceable and impossible to govern. Informal delivery may look fast until a key person leaves, a regulator asks for evidence or a dispute exposes that nobody agreed what "complete" meant.

But the case for process is not the same as a case for process supremacy. Controls are instruments. Once they become the definition of delivery, the programme starts managing the representation of work instead of the work itself.

3. A Composite Programme in 2007

Consider a composite financial-services programme replacing several ageing operational systems while introducing a common process across three business units. The programme is eighteen months old. Its monthly report is green-amber. Design is 96 per cent complete. Test preparation is on plan. Ninety-eight per cent of mandatory governance products have been approved.

At a working session, however, an operations manager demonstrates a recurring exception. A transaction rejected by the new rules must be re-keyed into a separate application. The re-keying takes four minutes, creates a second reference number and breaks the audit trail between the original instruction and the corrected record. At expected daily volumes, the workaround would require twenty additional staff.

The issue is real, repeatable and material. Yet it does not fit the established reporting route. The requirements document contains the stated rule. The design has been approved against that requirement. The test script confirms the system rejects the transaction correctly. Each artefact is compliant.

The operations manager asks whether the end-to-end process has been tested with realistic exception volumes. The programme team explains that volume testing covers technical throughput, while business acceptance covers functional outcomes. The gap between those definitions belongs to neither team.

A change request is raised. It is classified as a new requirement because the approved specification did not state that rejected transactions must retain a single reference. The change board defers it pending cost and schedule analysis. The monthly dashboard remains green-amber because no approved baseline has yet moved.

Nothing in this sequence is irrational. Every participant follows the agreed process. The process nevertheless converts operational evidence into a scope dispute and delays the decision until the problem is expensive.

The programme is compliant at every checkpoint and wrong in the space between them.

4. The Structural Mechanism

Four forces sustain this pattern.

1. Assurance is organised around artefacts.

Independent reviewers can inspect a document, approval or test result. They have less access to the everyday reality of users, hand-offs and exceptions. Artefacts become the practical unit of assurance, even when outcomes depend on interactions the artefacts divide.

2. Governance rewards variance management.

Programme reporting compares actual performance with an approved baseline. A newly discovered operational problem may not count as a variance until a formal change is accepted. This creates a period in which knowledge has changed but status has not.

3. Accountability follows organisational boundaries.

Technology owns system performance. Operations owns procedures. Change teams own training. Suppliers own contracted deliverables. The outcome depends on all four, but no single measure captures their combined effect. Each function can be compliant while the whole is unworkable.

4. Escalation carries political cost.

A leader who reports a process breach can point to a clear rule. A leader who reports that the programme is compliant but may still fail must challenge the adequacy of the governing model itself. That claim is harder to evidence and easier to dismiss as subjective.

These forces explain why the pattern persists across organisations. It is not caused primarily by timid individuals or excessive paperwork. It is produced by governance systems that give procedural evidence a recognised route to authority while operational doubt must fight for admission.

5. The Difference Between Control and Learning

Good programme control should make learning consequential. When evidence reveals that an assumption is false, governance should alter decisions, priorities or plans. The test of control is not whether the original method was obeyed. It is whether the programme can absorb new truth before that truth becomes failure.

This requires a distinction between three forms of evidence.

- Compliance evidence shows that required actions occurred.
- Delivery evidence shows that components work against defined criteria.
- Outcome evidence shows that the combined change works in the operating environment and advances the intended business result.

All three matter. The error is allowing the first to stand in for the other two.

A stage gate should therefore ask more than whether documents are complete. It should ask what has been learned since the previous gate, which assumptions have been invalidated and whether any formal success measure now conflicts with observed operating reality.

A risk register should do more than allocate owners. It should show what evidence would cause the programme to change course and who has authority to act when that evidence appears.

Testing should include end-to-end scenarios built around exceptions, hand-offs, peak conditions and workarounds, not merely the happy path represented in approved

requirements. The operational people who inherit the result must be able to demonstrate where the design fails, even when every component meets specification.

6. A More Useful Governance Contract

The answer is not to abandon process or celebrate improvisation. Large programmes need disciplined decisions, traceability and repeatable controls. The answer is to make the process explicitly subordinate to the outcome.

A practical governance contract can establish five rules.

1. Every major gate must include direct outcome evidence.

A signed pack is insufficient. Leaders should see a working demonstration, realistic scenario, operational rehearsal or measurable result appropriate to the stage.

2. Material evidence can reopen an approved decision.

Approval should close debate only until better evidence appears. Reopening a decision is not governance failure; refusing to reopen it may be.

3. Status reflects current knowledge, not baseline mechanics.

If credible evidence threatens the outcome, status changes immediately. The programme need not wait for a change request to be costed before acknowledging the risk.

4. Cross-boundary outcomes have named owners.

Where value depends on technology, process, people and supplier activity together, one accountable leader must own the combined result. Component owners remain responsible for their parts, but cannot collectively leave the whole unowned.

5. Assurance samples reality as well as records.

Reviewers should trace a small number of transactions, user journeys or business scenarios from beginning to end. A carefully chosen sample can reveal more about delivery than another review of document completeness.

These rules do not weaken control. They make control sensitive to the evidence that matters.

7. What Leaders Should Notice

The earliest warning sign is not a missing document. It is a repeated sentence: "That is outside the scope of this forum." When operational issues move between design, testing, change and commercial meetings without any one forum owning the outcome, process has begun to fragment reality.

A second warning sign is a dashboard that improves while workarounds multiply. Local teams often compensate for design defects long before formal measures deteriorate. Their effort protects the programme's reported status while transferring cost and risk into operations.

A third is the classification of inconvenient evidence as a new requirement. Sometimes it genuinely is. But when a programme repeatedly treats basic usability, exception handling or operational continuity as late scope, the original specification may have captured

components without capturing the job to be done.

Leaders should also be wary of unanimous approval. In a highly procedural environment, agreement may mean only that each function's conditions have been satisfied. It does not prove that the combined result is coherent.

8. Doing the Work

Process is valuable because complex delivery cannot depend on memory, personality or goodwill. It creates a common language for decisions and gives organisations the evidence to challenge, learn and account for their actions.

Yet process has no independent claim to success. Its value lies in helping people deliver a result under uncertainty. When adherence becomes the objective, the organisation can execute the method precisely while preserving the very problem the programme was meant to solve.

The governing question must remain plain: does the evidence show that the change will work, at the required scale, in the hands of the people who must use it?

If the answer is uncertain, another completed checklist is not progress. It is only proof that the checklist was completed.