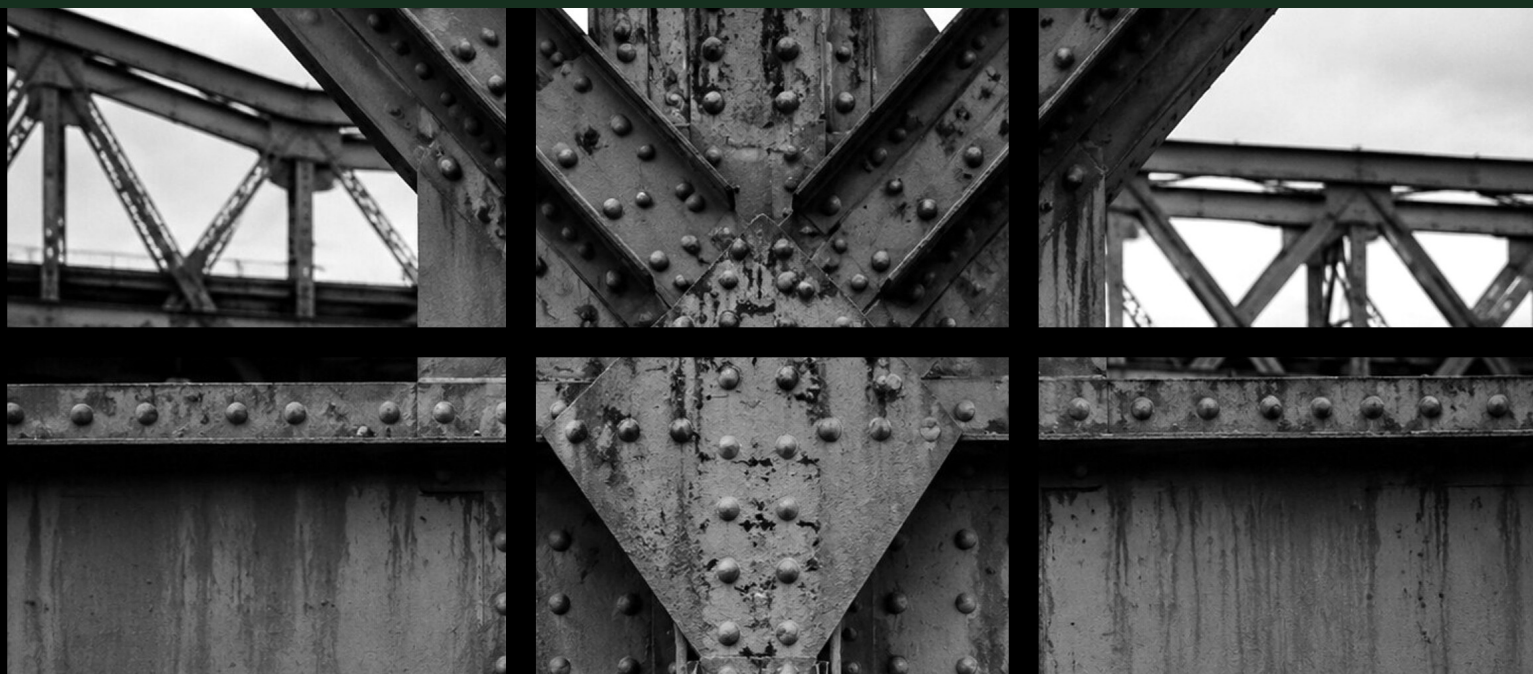


BEYOND TEMPLATE COMPLETION

The Template Is Becoming the Deliverable — and Programme Leaders Are Letting It

Why the paperwork boom around programme offices is obscuring whether the work is actually ready



Giovanni Leonardi

April 2007

The Template Is Becoming the Deliverable — and Programme Leaders Are Letting It

Programme · No. 77

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

“A completed template is evidence that a conversation was recorded, not that the problem was resolved.”

The Sentence That Should Worry Us

“The design is complete. The design document has been approved.”

That sentence is becoming common in programme meetings, and it is beginning to mean something quite different from what it appears to mean.

In one representative programme, a 64-page design pack had passed review, carried seven signatures and satisfied the required document checklist. Yet eleven sections still contained “to be confirmed”, two operating assumptions contradicted one another, and no manager had agreed who would own failed transactions after go-live. Testing was already booked. The deliverable was complete; the design was not.

This is not an isolated administrative irritation. Across large programmes in 2006 and 2007, templates are moving from being useful containers for judgement to becoming substitutes for judgement. The change is subtle because the paperwork looks like discipline.

Why the Template Is Winning

The rise of the modern programme office has brought real benefits. Common business cases, risk logs, stage-gate packs, requirements catalogues and change forms allow leaders to compare work that was previously described in incompatible ways. Outsourced delivery makes documentary clarity essential. Audit and regulatory scrutiny make an undocumented choice difficult to defend. A repeatable form can stop an enthusiastic team from skipping an awkward question.

The serious case for templates is therefore strong. Without them, programmes can become collections of private spreadsheets, verbal commitments and decisions that nobody can reconstruct six months later.

But the form is only useful while it remains subordinate to the decision it serves.

The failure mechanism begins when assurance tests whether each section is populated rather than whether the underlying uncertainty is resolved. Teams quickly learn the new definition of progress. An open question is converted into an assumption. A disagreement is moved to an appendix. A missing decision becomes an action with a due date safely beyond the next gate. The document can now advance even though the work cannot.

A completed template is evidence that a conversation was recorded, not that the problem was resolved.

Once this distinction is lost, three distortions follow:

- The artefact becomes the milestone. “Design complete” means the document was signed, not that the operating model is coherent.
- Review becomes editorial. Senior time is spent correcting wording and format while material trade-offs remain untouched.
- Accountability hides inside fields. An “owner” is entered because the cell requires one, although that person lacks the authority to decide.

The programme then develops two realities. In the reporting reality, deliverables move from draft to approved and traffic lights remain stable. In the delivery reality, unresolved choices reappear during build, test and transition, where they are more expensive and harder to reverse.

The Cost Arrives Later

Template culture is attractive because its cost is deferred.

The additional document takes two days now; the unresolved operating decision produces six weeks of rework later. The gate is passed on time; the test cycle absorbs the disagreement. The requirements catalogue reaches version 1.0; users discover that three individually approved requirements cannot coexist in the same process.

By then, the programme rarely blames the template. It blames poor execution, late business engagement or weak testing. The paperwork that allowed uncertainty to travel downstream escapes scrutiny because it was itself compliant.

When an artefact can be approved with a consequential question still unanswered, the approval process is certifying the document, not controlling the programme.

The remedy is not another mandatory section called “unresolved decisions”. That would repeat the error.

Instead, every major deliverable should be tested against the condition it claims to establish. A design is complete when the decisions needed for build and operation are made, bounded or explicitly accepted by someone with authority. A business case is complete when benefits, costs, risks and ownership form a decisionable proposition. A transition plan is complete when named operational managers can show how service will be received, supported and controlled.

The document records that condition. It does not create it.

What Leaders Should Ask Now

Programme leaders can expose template culture with a few plain questions:

- What can the team now do that it could not do before this document was approved?
- Which important uncertainty has been removed?
- What decision does the approval represent, and who is accountable for it?

- If the document disappeared tomorrow, what evidence would still show that the work was genuinely ready?

These questions are more demanding than checking completeness because they require judgement. They may reveal that a beautifully presented pack is not ready for approval. They may also show that a short, imperfect paper contains enough evidence for a sound decision.

That is precisely the point.

The programme discipline now spreading through organisations will mature only if it distinguishes standardisation from surrender. Templates should make good thinking visible, transferable and reviewable. They should not define thinking as the successful completion of boxes.

We should be uneasy whenever a meeting celebrates the approval of a deliverable without being able to say what changed in the real work. The document is not the outcome. In too many programmes, we are beginning to forget the difference.