

BEYOND THE TEMPLATE

Template Culture Makes Deliverables Look Like Delivery

How document-driven programmes mistake completed products for resolved decisions



Template Culture Makes Deliverables Look Like Delivery

Programme · No. 86

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

“The paperwork had divided the question more efficiently than the organisation had answered it.”

The Pack Was Complete

The programme office had prepared twenty-six required products for the design gate. The combined pack ran to 412 pages. Every template carried a version number, an owner, a reviewer and an approval box. The steering committee received it five working days before the meeting.

Within twelve minutes, the gate was approved.

Three weeks later, the operations lead discovered that no one had decided who would own the customer records rejected during migration. The data plan described extraction, cleansing, mapping, loading and reconciliation. The operating model described future roles. The risk register mentioned data quality. All three documents were complete. The decision connecting them did not exist.

This is how template culture defeats delivery. It does not usually produce missing paperwork. It produces an abundance of paperwork that allows a programme to look governed while its decisive uncertainties remain unresolved.

The problem is not the template itself. The problem is the quiet promotion of the deliverable from a means of thinking to a substitute for thought.

Why Templates Took Hold

The appeal is easy to understand. As programme management has become more formal, organisations have wanted consistency across workstreams, suppliers and business units. A standard business case, requirements catalogue, risk register and stage-gate pack create a common language. They make assurance easier. They reduce dependence on individual memory. They allow a programme office to identify obvious omissions before those omissions become expensive.

That is the strongest case for template discipline, and it is a good one. A complex programme run entirely through conversation and personal judgement will become untraceable. When people leave, their reasoning leaves with them. When suppliers disagree, nobody can establish what was approved. When a steering committee asks why a decision was made, memory becomes advocacy.

Templates are therefore valuable controls. But controls become dangerous when they start determining what counts as reality.

A template has fixed fields because fixed fields are easy to review. Delivery does not have fixed fields. The most important issue may sit between workstreams, emerge as an exception, or invalidate an assumption that no standard section asks anyone to revisit.

The document then exerts a subtle pressure: if an issue does not fit the template, it is treated as incomplete analysis rather than inconvenient truth.

How the Substitution Happens

I have seen a recognisable sequence repeat across programmes.

1. The method defines mandatory products. Each workstream is judged on whether those products are complete.
2. The programme office reports product completion. A simple red-amber-green measure turns document production into visible progress.
3. Reviewers focus on conformance. They can check whether sections are populated more quickly than they can challenge whether the underlying decision is sound.
4. Authors learn what secures approval. They write to satisfy the reviewer and avoid reopening settled scope.
5. The approved document becomes evidence that the subject is controlled. Later operational concerns are classified as changes because the relevant product has already passed its gate.

No participant needs to be careless for this to occur. In fact, diligence can make it worse. The more conscientiously people complete the prescribed material, the more confidence the programme places in the completeness of its view.

In the composite migration example, the data manager owned the conversion plan, the operations lead owned the future process and the supplier owned the loading routine. Each template was internally coherent. None required a named decision on rejected records at the boundary between them.

The missing decision appeared only when a trial conversion rejected 8,400 of 160,000 customer records. The programme could explain the rejection rate, but not who would investigate the records, which system would hold them, how long resolution would take or whether go-live could proceed with an unresolved balance.

The paperwork had divided the question more efficiently than the organisation had answered it.

A deliverable is complete only when it has forced the decisions that its existence is meant to evidence.

What Textbooks Prescribe and Programmes Practise

Textbooks tend to present a clean chain: define the product, assign responsibility, produce it, review it, approve it and control subsequent change. In that chain, the document is a faithful representation of prior reasoning.

Living programmes often reverse the sequence. The deadline for the gate arrives first. The product must therefore be produced. Analysis is compressed to fit the required structure. Review comments concentrate on what can be corrected before the meeting. Approval closes the document. The closed document then limits which questions may be asked without raising a change.

The difference is not between method and chaos. It is between documentation that records judgement and documentation that manufactures the appearance of judgement.

Three tests expose the difference.

- Decision test: What decision will this deliverable enable, and who must make it?
- Evidence test: What observation, trial, figure or exception could prove its central assumption wrong?
- Boundary test: Which other product, workstream or role must agree before this one is genuinely complete?

If a document has no clear answer to those questions, its approval adds administrative certainty without reducing delivery uncertainty.

The Programme Office Must Change Its Question

The programme office is often blamed for template culture, but it is more accurately its transmission mechanism. It uses the measures leaders ask for and enforces the method the organisation has selected.

The remedy is not to abolish standard products. It is to change what completion means.

A gate pack should identify unresolved decisions before it lists completed documents. A product review should begin with assumptions and evidence, not formatting and required headings. A workstream should be able to declare a document incomplete because a cross-boundary decision remains open without being punished for poor compliance.

Most importantly, the programme office should stop asking, "Has the deliverable been approved?" as though that settled the matter. It should ask:

- What changed because this work was done?
- Which risk is now better understood?
- Which decision has become possible?
- What remains uncertain despite the approval?
- What operational evidence supports the conclusion?

These questions restore the proper relationship. The template serves the decision; the decision does not serve the template.

Delivery Is Not a Document Set

The temptation to equate deliverables with delivery will remain strong. Documents are visible, countable and contractually useful. Real progress is uneven, contested and sometimes embarrassing. A programme can report that every product is on schedule. It is harder to report that a fundamental operating question remains unanswered.

Yet that discomfort is precisely what governance is for. Its purpose is not to make uncertainty disappear from the pack. It is to make uncertainty visible early enough for leaders to act.

We should keep the templates. We should keep the gates, the audit trail and the discipline of written decisions. But we should refuse the comforting fiction that a completed set of documents is a completed piece of work.

The value of a deliverable lies in the judgement it captures, the evidence it exposes and the decision it enables. Without those, the programme has not delivered. It has merely filed.