

BEYOND THE DELIVERABLE

Capability Maturity — Building the Muscle That Survives the Programme

A Practitioner's Perspective on Sustainable Organisational Capability in Programme-Driven Environments



Giovanni Leonardi

January 2008

Contents

1. Executive Summary	3
2. The Temporary Structure and the Permanent Need	3
3. Why Programmes Are Structurally Hostile to Capability Building	4
4. The Capability Gap in Practice	5
5. What Distinguishes Programmes That Build Lasting Capability	5
6. The Maturity Dimension	6
7. The Organisational Absorptive Capacity Problem	7
8. Building the Muscle – A Different Approach	7
9. The Leadership Dimension	8
10. The Uncomfortable Truth	8

Capability Maturity – Building the Muscle That Survives the Programme

Programme · No. 107

Originally written January 2008 | Re-edited and re-rendered for the WhereBeyond Edition, 2025

“The programme that builds nothing but its own outputs has, in a meaningful sense, failed – no matter how many milestones it hits on time.”

1. Executive Summary

Programmes are, by their nature, temporary structures. They are established to deliver a defined set of outcomes and then, in theory, to dissolve – leaving behind not only those outcomes but the organisational capability to sustain and build upon them. In practice, this second objective is almost universally neglected. The programme delivers its milestones, declares success, and disbands, only for the organisation to discover months later that it cannot operate what was built, cannot maintain what was changed, and cannot repeat what was achieved.

This essay explores why this pattern persists. It argues that the root cause is not negligence but a structural misalignment between how programmes are designed, governed, and incentivised and what genuine capability maturity requires. The programme's temporary nature – its greatest structural advantage for driving change – is simultaneously its greatest liability for embedding it. The disciplines of programme management, as currently practised, optimise for delivery at the expense of durability.

The essay traces the capability gap through its most common manifestations: the knowledge that walks out when contractors leave, the processes that revert within weeks of go-live, the skills that were never transferred because the programme moved too fast to teach. It then examines what distinguishes the rare programmes that do leave lasting capability behind, and proposes that capability maturity must be designed into the programme from the outset – not bolted on as a transition activity in the final phase.

2. The Temporary Structure and the Permanent Need

There is an inherent tension at the heart of every programme that aims to transform an organisation. The programme itself is temporary – a bounded intervention with a start date, an end date, and a mandate to deliver specific outcomes within that window. But the capability it seeks to build, the ways of working it seeks to embed, the organisational muscle it seeks to develop – these must be permanent, or at least durable enough to outlast the programme by years.

This tension is not new, and most programme managers would acknowledge it if asked. Yet the way programmes are structured, funded, and governed consistently prioritises the temporary over the permanent. The programme board reviews milestones, not capability transfer. The programme plan sequences deliverables, not learning curves. The programme budget funds delivery resources, not the patient, repetitive work of

embedding new skills in permanent staff.

The result is predictable. The programme delivers – often impressively – and then departs. What remains is a set of outputs: new systems, new processes, new organisational structures. What does not remain, in most cases, is the organisational ability to operate, maintain, evolve, and improve those outputs without the programme's scaffolding.

In my experience, this is not a problem of awareness. Senior leaders understand, at least in principle, that capability must outlast the programme. The problem is structural. The programme's incentive system, its governance model, its resourcing approach, and its definition of success all conspire against the patient, unglamorous work of building durable capability.

3. Why Programmes Are Structurally Hostile to Capability Building

To understand why capability maturity is so consistently neglected, it helps to examine the structural features of programmes that work against it.

The milestone tyranny. Programmes are governed by milestones, and milestones measure delivery, not adoption. A milestone might read "new process deployed" or "system go-live complete," but it will rarely read "permanent staff demonstrably capable of operating the new process without programme support." The difference matters enormously. The first can be achieved by the programme team alone; the second requires sustained engagement with the operational organisation, and that engagement is expensive, slow, and difficult to plan with precision.

The contractor dependency. Many programmes rely heavily on external contractors and consultants who bring specialist skills the organisation does not possess. This is often necessary and appropriate. But it creates a structural problem for capability transfer: the people who know how to do the work are temporary, and their departure is built into the programme plan. Knowledge transfer is typically scheduled as a late-stage activity – a few weeks of documentation and shadowing before the contractors leave. In practice, this is almost never sufficient. The tacit knowledge, the judgment calls, the workarounds and contextual understanding that make complex work possible – these cannot be transferred in a handover document.

The pace problem. Programmes move fast, and they are designed to move fast. Speed is a virtue in programme delivery: it maintains momentum, holds stakeholder attention, and reduces the window of exposure to external risks. But capability building is inherently slow. Learning new skills, developing new habits, building confidence in new ways of working – these take time, repetition, and the freedom to make mistakes. The programme's pace is fundamentally incompatible with the learner's pace, and when the two conflict, the programme's pace invariably wins.

The success definition. Perhaps most fundamentally, programmes define success in terms of their own outputs, not in terms of the organisation's subsequent performance. A programme that delivers on time and on budget is deemed successful, even if the organisation struggles to operate what was delivered. By the time the capability gap becomes visible – typically six to twelve months after the programme closes – the programme team has dispersed, the programme board has dissolved, and there is no structure left to be held accountable.

4. The Capability Gap in Practice

The consequences of this structural misalignment are visible across sectors and across programme types. The pattern varies in its specifics but is remarkably consistent in its shape.

The knowledge walkout. When a programme closes and its contractors depart, they take with them not only their specialist skills but their understanding of why decisions were made, how components interact, and where the fragile points lie. The permanent staff who inherit the programme's outputs often find themselves operating systems and processes they do not fully understand. They can follow the documented procedures, but they cannot diagnose problems, adapt to changing circumstances, or make the judgment calls that the programme team made instinctively.

The process reversion. New processes introduced by a programme frequently revert to older ways of working within weeks or months of go-live. This is not sabotage; it is the natural result of insufficient embedding. The permanent staff were trained on the new process but never had the opportunity to practise it under realistic conditions with adequate support. When the programme's scaffolding is removed and real operational pressures reassert themselves, the path of least resistance leads back to the familiar.

The improvement plateau. Even where new capabilities are initially adopted, they often plateau at a basic level of competence. The organisation can operate the new system or follow the new process, but it cannot optimise, refine, or evolve it. The capability was transferred as a set of procedures rather than as a genuine competence, and without the deeper understanding that would enable continuous improvement, the organisation is stuck at whatever level of performance it reached during the programme's final weeks.

The repeat commissioning cycle. Perhaps the most telling symptom is what happens next. Unable to sustain or develop the capability the programme was supposed to build, the organisation commissions another programme – or another set of consultants – to address the same underlying need. The cycle repeats, each iteration consuming resources and management attention without building the lasting muscle that would break the pattern.

5. What Distinguishes Programmes That Build Lasting Capability

The pattern described above is common, but it is not universal. In my experience, a minority of programmes do succeed in building capability that endures. What distinguishes them is not a single practice or technique but a fundamentally different orientation – one that treats capability maturity as a primary objective of the programme, not a secondary consequence of delivery.

Several characteristics recur in programmes that build lasting capability:

Capability is designed in, not bolted on. The most effective programmes treat capability transfer not as a transition activity but as a design principle. From the outset, the programme is structured so that permanent staff are involved in delivery – not as observers or trainees, but as participants who carry increasing responsibility as the programme progresses. This is slower and more expensive than having the programme team deliver everything, but it produces capability that survives the programme's departure.

The programme invests in the permanent organisation's leaders. Capability does not sustain itself; it requires leadership that understands it, values it, and actively maintains it. Programmes that build lasting capability invest heavily in developing the operational leaders who will own the capability after the programme closes. This means more than briefing them on what was delivered; it means involving them in the decisions that shaped the design, so they understand not just what the new capability is but why it is that way.

Success is measured after the programme closes. The programmes that build durable capability define success partly in terms of the organisation's performance in the months after the programme ends. This requires a governance mechanism that survives the programme – a benefits realisation process, a post-programme review, or an explicit handover to an operational governance forum that will monitor capability maturation.

The pace is deliberately managed. Rather than driving for the fastest possible delivery, these programmes pace themselves to the organisation's absorptive capacity. They build in time for permanent staff to practise new skills, make mistakes, and develop confidence before the programme's support is withdrawn. This often means extending the programme timeline or phasing the withdrawal of programme resources rather than executing a clean handover on a fixed date.

Knowledge transfer is continuous, not terminal. Instead of scheduling knowledge transfer as a final-phase activity, effective programmes embed it throughout the delivery lifecycle. Every design decision, every implementation choice, every problem solved becomes an opportunity for permanent staff to learn – not from a document, but from participation in the work itself.

6. The Maturity Dimension

Capability maturity is not a binary state – an organisation does not simply "have" or "lack" a capability. Maturity develops through stages, and understanding these stages is essential for programmes that aim to build lasting capability.

At the most basic level, capability is procedural: the organisation can follow a defined process to produce a defined output. This is typically where programmes leave the organisation – with documented procedures and staff trained to follow them. It is necessary but far from sufficient.

The next level is competent: the organisation can not only follow the process but adapt it to varying circumstances, diagnose problems when they arise, and make informed trade-off decisions. This requires deeper understanding and more extensive practice than most programmes provide.

Beyond competence lies mastery: the organisation can optimise the capability, innovate within it, and teach it to others. This level is rarely achieved during a programme's lifetime, but the programme can create the conditions that make it achievable – by building sufficient depth of understanding, by establishing communities of practice, and by ensuring that the operational leadership has both the mandate and the skill to drive continuous improvement.

The pattern I have observed is that programmes consistently aim for procedural capability and declare success when they achieve it. But procedural capability is fragile – it depends on stable conditions and breaks down under pressure or change. Competent capability is resilient, and mastery is generative. The gap between where programmes leave the organisation and where it needs to be is, in most cases, at least one full maturity level.

7. The Organisational Absorptive Capacity Problem

One of the least discussed factors in capability maturity is the organisation's absorptive capacity – its ability to take in, process, and embed new capabilities while continuing to operate. This capacity is finite, and programmes routinely exceed it.

The problem is compounded when multiple programmes run concurrently, each demanding that the operational organisation absorb new ways of working. The permanent staff who must learn new skills, adopt new processes, and adapt to new systems are the same people who must continue to deliver operational performance. They cannot stop the engine to rebuild it; they must rebuild it while it runs.

Programmes that recognise this constraint plan accordingly. They sequence capability demands so that the organisation is not overwhelmed. They identify the key individuals whose adoption is critical and protect their time for learning. They monitor absorption indicators – not just training completion rates, but actual behavioural change, confidence levels, and the ability to perform new tasks without support.

Programmes that ignore absorptive capacity – which is to say, most programmes – create a familiar pattern: training is delivered, boxes are ticked, and the organisation appears ready. But when the programme departs and the support structure is removed, the reality becomes apparent. The organisation was trained but not capable, briefed but not competent, informed but not transformed.

8. Building the Muscle – A Different Approach

If capability maturity is to be taken seriously, it requires a different approach to programme design – one that treats the building of permanent organisational muscle as a first-order objective alongside the delivery of programme outputs.

This approach has several implications:

The programme's purpose is not to deliver change but to build the organisation's ability to deliver and sustain change. The outputs are necessary but not sufficient; the capability to operate, maintain, and evolve those outputs is the real measure of success.

Resourcing must reflect the dual objective. If the programme is to build capability as well as deliver outputs, it needs resources dedicated to both. This means budgeting for embedded coaching, for extended transition periods, for the slower pace that genuine capability transfer demands. It also means resisting the pressure to cut these resources when the programme faces cost or time pressure – which it inevitably will.

Governance must track capability, not just delivery. The programme board needs visibility of capability maturity alongside the traditional measures of schedule, cost, and scope. This requires new metrics – not just "training delivered" but "capability demonstrated," measured through practical assessment, operational performance, and the organisation's ability to operate without programme support.

The exit criterion must change. A programme should not close until the organisation has demonstrated – not just claimed – that it can sustain the new capability without the

programme's scaffolding. This is a higher bar than most programmes set, and it will sometimes mean extending programme timelines. But the alternative – closing on schedule and watching the capability erode – is more expensive in the long run.

The programme must build its own redundancy. The ultimate test of a capability-building programme is whether the organisation needs the programme. A programme that is still essential to the operation of the capability it delivered has not succeeded; it has created dependency. The programme's goal should be to make itself unnecessary – to build capability so thoroughly embedded that the programme's departure is barely noticed.

9. The Leadership Dimension

Capability maturity is ultimately a leadership challenge, not a programme management challenge. The programme can create the conditions for capability to develop, but only the organisation's permanent leadership can sustain it.

This means that capability-building programmes must invest in leadership development as a core activity, not a peripheral one. The operational leaders who will own the capability after the programme closes need to understand not just what was delivered but why, not just how the new processes work but what principles underpin them, not just what the programme achieved but what it learned along the way.

The leader who inherits a capability without understanding its design rationale is condemned to maintain it as a rigid procedure, unable to adapt it when circumstances change – which they always do.

In my experience, the programmes that build lasting capability are the ones where the operational leadership is actively involved in the programme's decision-making, not just informed of its decisions. When leaders participate in design choices, trade-off discussions, and problem-solving, they develop the understanding necessary to steward the capability through its post-programme evolution.

10. The Uncomfortable Truth

The pattern described in this essay persists because it serves certain interests. Programmes that deliver quickly and cleanly, without the messy and expensive work of genuine capability transfer, look good on paper. They come in on time and on budget. Their programme managers are rewarded and move on to the next assignment. The capability gap that follows is someone else's problem – usually the operational manager who inherits an organisation that has been changed but not strengthened.

The consulting and contracting industry, too, has limited incentive to solve this problem. An organisation that builds genuine internal capability is an organisation that needs fewer consultants. The repeat commissioning cycle described earlier is, from the supplier's perspective, a reliable revenue stream.

Breaking this pattern requires organisations to demand more of their programmes – and to be willing to invest more in them. A programme that builds lasting capability will cost more upfront, take longer, and produce fewer impressive milestones in its early phases. But it will leave behind something that the faster, cheaper alternative does not: an

organisation that is genuinely stronger than it was before.

The programme that builds nothing but its own outputs has, in a meaningful sense, failed – no matter how many milestones it hits on time. The programme that builds the muscle to sustain, adapt, and improve those outputs has succeeded in the only way that ultimately matters. The distinction is uncomfortable, because it challenges the metrics by which most programmes – and most programme managers – are judged. But it is the distinction on which lasting organisational transformation depends.