

WHY PROGRAMMES FAIL - AND WHAT RECOVERY REQUIRES

Averruncus Solutions Ltd

October 2026

EXECUTIVE SUMMARY

Most large programmes fail. The Standish Group's CHAOS Report, which has tracked project outcomes for decades, records that only around 31% of projects succeed in meeting time, budget and scope goals. Close to half are "challenged" - late, over budget, or missing key features. Around one in five fail outright.

The failure rate has not improved materially in a decade, despite the widespread adoption of Agile, better tools and more mature delivery frameworks. This suggests the problem is not methodology, but something more fundamental.

This paper argues that programme failure is not primarily a technical problem, but rather a governance and behavioural problem. Programmes fail because assumptions are rarely validated, dependencies aren't effectively managed, reporting reflects optimism rather than reality, and the people best placed to challenge the plan are unable to do so.

Five conclusions shape what follows:

- **The causes of failure are known and repetitive.** The evidence base is extensive and consistent. Yet organisations continue to repeat the same mistakes.
 - **Failure is usually visible internally long before it becomes visible externally.** The problem is not detection. It is openness and escalation.
 - **The governance structures meant to surface problems often suppress them.** Optimism bias and the incentive to report good news are structurally embedded.
 - **Recovery requires a different set of skills from delivery.** It requires diagnosis, honesty, and the willingness to stop something if it is not working.
 - **The most useful intervention is often external.** The people inside a failing programme are frequently the least able to see it clearly for what it is.
-

1. INTRODUCTION

Large programmes and projects fail at a rate that would be unacceptable in any other area of business. The Standish Group's CHAOS Report consistently finds that fewer than one in three projects succeed in meeting their original objectives. Roughly half are

"challenged" - delivered late, over budget, or with reduced functionality. Around 19% fail outright.

The failure rate has remained remarkably stable. A longitudinal study published in the PM World Journal compared data from 2017 with data from 2020–2024 and found "stagnation in success rates over the last decade," concluding that "the widespread adoption of Agile has not resulted in a proportional increase in outright success."

The McKinsey research commonly cited in transformation literature puts the failure rate at around 70%. The UK National Audit Office has repeatedly documented over-optimism in government projects, noting that "optimism bias in public sector projects is not a new phenomenon. But it persists, frequently undermining projects' value for money as time and cost are underestimated and benefits overestimated."

This is not a technology problem. It is not a methodology problem. It is a problem of how organisations plan, govern and challenge their most important work.

This paper examines why programmes fail, drawing on public evidence and observed practice. It then sets out what recovery requires.

2. WHY PROGRAMMES FAIL

The evidence is consistent

Public inquiries, academic research and consulting studies converge on the same conclusions. The specifics vary, but several recurring patterns emerge.

Optimism bias is systemic. The National Audit Office has documented over-optimism in respect of project outcomes across decades of government projects, identifying "project complexity and an organisation's culture of challenge" as contributory factors.

Governance fails to challenge. A study of an IT project failure found that "project managers, having fallen into a subservient role to the executive, didn't challenge the decision" and instead pushed unrealistic timelines onto suppliers. The result was a programme that appeared to be progressing while delivering nothing of value.

Reporting reflects what people want to hear. The NAO's work on over-optimism notes that "risks and issues [were] not identified in key messages but set out in the detail of reporting and further obfuscated by overly optimistic mitigations." The information existed, but it was buried.

Business cases are not revisited. Research and practice suggest organisations rarely review business cases after approval. When circumstances change, nobody checks whether the original rationale still holds.

Junior people are given responsibility without capability. Smaller projects are often assigned to people without the training or experience the role requires. The same pattern applies at scale, where complex programmes are led by people who lack the standing to challenge the sponsor or the experience to recognise when a plan is unravelling.

The structural problem

A programme manager's role finishes when the programme ends, and so continuing is almost always the safer option for them. Stopping requires confronting the sponsor, admitting that money has been wasted, and accepting that the original case may have been wrong. The incentives only point in one direction.

Meanwhile, the sponsor - often a senior executive with many other responsibilities – tends only to see what the programme reports. If the reporting is optimistic, the sponsor has no reason to doubt it. The governance structure, which should provide independent challenge, frequently becomes a mechanism for confirming what the programme already believes.

The result is a system that suppresses bad news until it can no longer be suppressed. By the time the truth emerges, the cost of recovery is far higher than it would have been if the problem had been surfaced earlier.

The problem with green

Programme status reporting has a well-known distortion, and it shapes behaviour throughout a programme. Red attracts scrutiny, additional governance and reputational damage for the sponsor. Amber invites pressure to explain when it will turn green. Green is the path of least resistance, even when it is not accurate.

The consequence is that programmes that should be amber remain green for long periods. Risks that have not yet been mitigated are reported as though they have been. The status becomes a measure of what the organisation wants to hear rather than what is true.

This is not dishonesty at an individual level. It is a rational response to the incentives in the system. If the organisation punishes red, treats amber as a problem to be resolved and rewards green, then green becomes the only safe answer. The organisation's risk appetite shapes this - some treat an unresolved risk as a normal state to be managed, others as a failure demanding immediate resolution. Where the latter prevails, honest reporting becomes a liability for those doing it, accelerating the drift toward green.

The corollary is that a programme which has been amber for six months may be better managed than one which has been green throughout. Amber reflects an honest assessment of uncertainty. Green often reflects its suppression.

The human dimension

The human dimension matters too. Research by INSEAD found that leadership responses to underperformance were often "a defensive reaction to leaders' own anxiety - a way of avoiding true self-reflection by making the organisation the problem." Programmes are shaped by anxiety, ambition and the fear of being found wanting. When the stakes are high, the incentive to report success rather than admit failure is powerful.

3. HOW FAILURE BECOMES VISIBLE

Programmes in difficulty rarely surface the problem themselves. The truth typically emerges through an external trigger:

- **Regulatory scrutiny.** A regulator asks a question the organisation cannot answer. The gap between what is reported and what is true becomes visible.
- **Audit findings.** Internal or external audit identifies a control failure or finds that reported progress cannot be evidenced.
- **A new senior appointment.** A senior leader joins, or moves into a role, without the investment in the existing narrative that the incumbents have. They ask the obvious questions, and the answers do not add up.
- **A crisis.** A material operational failure or client escalation brings the problem into the public domain.

The common thread is that the organisation's own governance failed to surface the issue. The problem was known - or knowable - internally, but it was not escalated. The information existed in the detail of reporting, or in the experience of the people doing the work, but it did not reach the people with the power to act.

The later the problem is discovered, the more expensive recovery becomes. A programme that is six months from completion and has spent 80% of its budget has far fewer options than one that is two years out. The ability to change course, re-scope, or build a fallback diminishes as time and money are consumed.

4. MANAGING DEPENDENCIES AND ASSUMPTIONS

Most failing programmes are in such a state because of something they do not control. This could be either a dependency - on another programme, a supplier, a regulator, or a system that is not delivering. Or it could be an assumption, one that was made early and never validated.

These two things - dependencies and assumptions - are frequently cited as significant sources of programme failure, and they are among the least well managed in practice. Dependencies are usually documented as part of a project; assumptions usually only to support the business case. Frequently, neither is followed up.

Assumptions

Assumptions are the things a business case rests on that nobody has tested. A delivery estimate built on a comparison to previous work, an adoption rate taken from a similar project, a cost based on a supplier's initial quote. Whilst these may appear reasonable at the time, they become dangerous when they are never revisited.

The discipline is straightforward, though rarely applied:

- **Document the assumptions** when the business case is written, not after, but follow this through to the programme documentation.
- **Classify them by uncertainty** - from highly uncertain to very certain - so that the ones which matter most are the ones watched most closely.
- **Set a validation date for each assumption** and make those dates milestones on the programme plan. An assumption that matters should be tested before the point at which the programme depends on it being true.
- **Track validation as the programme progresses.** An assumption is not a fact. It is a hypothesis that should be confirmed or disproved with evidence.
- **Be prepared to pause work.** If a key assumption has not been validated, continuing may mean that work is proceeding at risk and may be wasted effort should the assumption be disproved later.
- **Re-test the business case.** If the assumptions have changed, the case may no longer hold. That is a decision for the sponsor, but it should be a decision, not a silent drift.

Placing validation dates in the plan changes their status. An assumption with no date is merely a good intention. An assumption with a date is a deliverable, and the sponsor can see whether it has been tested by the point at which it matters. If the assumption cannot be validated by that date, that is itself the finding - and it should trigger the same conversation as any other missed milestone.

Dependencies

Dependencies are harder, because they involve another party whose priorities are not the programme's.

The typical pattern is well established. At the design stage, the solution identifies that part of the requirements will be delivered by a dependent project. Resources are not

allocated to those requirements, because they belong to someone else. That is rational at the time, and it is where the problem starts.

What follows is usually passive tracking. The programme invites the dependent project to its governance meetings. Progress against the requirements is monitored. But the programme cannot control delivery, only observe it. If the requirements are not a priority for the other team, they will drift, and the programme will watch it happen without being able to intervene.

There are three stages to managing this properly.

- **First, recognise the dependency as a risk, not a given.** A dependency on a higher-profile programme is not a plan. It is an exposure. The louder the programme, the less likely it is to prioritise work that only matters to someone else. If you cannot influence it, you need a fallback solution, and you need to cost it. Ideally, ensure that the specific dependency is reflected as requirements within the dependent programme, allowing trackability.
- **Second, escalate early and with evidence.** When the requirements are off track, ask the dependent project how they plan to recover. If no credible plan is forthcoming, escalate. The escalation should carry a clear recommendation, not just a problem. If escalation has no effect - if the timeframe remains unclear and progress cannot be evidenced - the fallback solution becomes the primary one.
- **Third, and most importantly, be in the room.** The ideal way to manage a dependency is for your programme to be a stakeholder of the programme you depend on. Attend their steering meetings. Confirm that the relevant requirements are being worked on. Challenge them about your interests. Understand their constraints and competing demands, so that you can make a case that fits their priorities rather than simply asserting your own.

This is work, and many programmes see it as intrusive. The dependent project does not want another programme at its steering meetings asking awkward questions. The default is passive tracking, which is where dependencies fail.

But the organisation benefits from the efficiency. A dependency that is visible and governed is cheaper to manage than one that fails late. Senior leadership should be clear that this is how dependencies are expected to be managed and should support programmes in managing them effectively.

The alternative is that the programme reaches the point where it can no longer proceed without the dependency, the dependency has not delivered, and the fallback has to be invoked under pressure, often using a tactical solution that lacks appropriate controls and efficiency. That is a decision made in a crisis rather than a change of direction made in advance. The difference in cost and disruption can be substantial.

Why amber becomes permanent

Not all risks can be closed by the programme. Some - economic conditions, regulatory change, the decisions of a parent organisation - have to be accepted and managed rather than resolved. Such risks should be identified separately from those that can be addressed. But they are rarely the ones keeping a programme amber.

These risks are often documented but not resolved for a reason. A programme can control its own delivery activity, and doing what it can control - delivering, reporting, governing - feels like progress. Addressing what it cannot directly control requires effort, diplomacy and the risk of an uncomfortable conversation. So, the risks stay open, the programme stays amber, and amber becomes permanent rather than transitional.

This connects to the earlier point about status reporting. A programme that has been amber for six months may accurately reflect a set of unresolved risks. The question to ask is "what has been done to address the risks?" If the answer is "they are being tracked," then the programme has not yet done the work.

A programme becomes genuinely green not when the risks have been logged and monitored, but when they have genuinely been addressed. That is what turns amber into green, and it takes effort and collaboration rather than process.

5. WHAT RECOVERY REQUIRES

Diagnosis before action

The instinct when a programme is in difficulty is to implement the fix as soon as possible. That instinct is usually wrong.

The first requirement is diagnosis. What is actually wrong? Is the plan unrealistic? Are the dependencies unmanageable? Is the problem technical, organisational, or both? Is the solution viable, or does it need to be replaced?

What should happen at this point is to check the programme's velocity. There is a danger of work being wasted and costs incurred if elements of the programme continue unchecked. Positive messaging is important to ensure the team is aware of the process and to reassure them so that key resources are not lost.

Recovery begins with a thorough assessment: review of documentation, interviews with the people doing the work, interviews with management, breakdown of the process, identification of assumptions, and an honest assessment of whether the original approach can be salvaged.

This is uncomfortable work. It often confirms what people already suspected but have not been able to say. The value is in making the problem explicit and testable, so that solutions can be evaluated against evidence rather than opinion.

The scope problem

One of the most common findings in a programme that has gone off track is that the scope has expanded while the original timelines and budget have remained. This is rarely a single decision; rather, it accumulates through added requirements, dependencies and regulatory change, each defensible in isolation. The original business case is rarely revisited, so the programme continues to be measured against constraints that no longer match the work.

Recovery requires confronting this directly. The first question is not "how do we deliver on time and on budget?" It is "what is the true scope, and what would it take to deliver it?" Once that is clear, the choice becomes explicit: reduce scope, extend time, increase cost, or some combination. The one option that is not available is to continue pretending the original constraints still apply.

Rebuilding governance

A programme in difficulty has often lost the governance structures that would have caught the problem. Reporting is optimistic. Risks are buried. Issues are not escalated.

Recovery requires rebuilding governance from the ground up:

- **A single source of truth.** A transparent, always-available view of status, risks and decisions that everyone can see.
- **Honest reporting.** Reporting that reflects reality, not the plan. If the programme is behind, say so.
- **Clear ownership.** Every workstream, every risk, every dependency has a named owner who is accountable.
- **Regular challenge.** Governance forums that test assumptions rather than confirm them.
- **Calibrated status.** A culture in which amber is an acceptable and honest state, and green is only used when the work is not only on track, but also that key risks have been addressed.

Decision-making under pressure

When a programme is in difficulty, decisions need to be made quickly and with incomplete information. The governance structures that work in normal times often fail under pressure. They become slow, risk-averse, and focused on blame rather than resolution.

Effective recovery requires a different approach:

- **Pre-agreed contingent decisions.** Secure approval for a fallback in advance, with a defined trigger. When the trigger arrives, the decision is already made, reducing the need for debate and re-planning.
- **Clear escalation.** The route of escalation is unambiguous. Issues that cannot be resolved at one level are escalated immediately, with a recommendation for action, not just a problem statement.
- **Authority to act.** The recovery lead needs the authority to make decisions, reallocate resources and change direction without seeking approval for every step.

The human element

Recovery is as much about people as it is about plans. A programme in difficulty has usually damaged trust - between the programme and the business, between the sponsor and the delivery team, between the organisation and its stakeholders.

Rebuilding trust requires honesty, visible progress and relationship repair. Delivering something tangible that demonstrates the programme can work is usually the fastest way to rebuild confidence.

6. WHY EXTERNAL HELP WORKS

The INSEAD research noted that "deliberate intervention from outside is often more effective at breaking the cycle" of defensive organising. This is a familiar observation to anyone who has worked in recovery.

The reasons are structural, not personal:

- **External people can see what insiders cannot.** The people inside a failing programme have usually normalised its problems. They have adapted to the dysfunction. An external perspective cuts through this.
- **External people can say what insiders cannot.** A programme manager who challenges the sponsor risks their job. An external recovery specialist does not.
- **External people are not invested in the original plan.** Insiders have often spent years defending a particular approach. An outsider can assess it on its merits.
- **External people can act as an honest broker.** When trust has broken down between the programme and the business, an external party can facilitate rebuilding it.

There is also a practical dimension. Advisors who work regularly on failing programmes have seen the pattern before. They recognise the symptoms, the reporting that doesn't reconcile, the risks that have been open for months, the dependency everyone is waiting on. They can often identify the actions required to get the programme back on track faster than the programme can identify them itself. Not because they are cleverer, but because they have seen it before and are not carrying the weight of the original plan.

For a programme under time pressure - which is most of them by the time external help is sought – such speed matters. The window in which recovery is achievable narrows as time and money are consumed. An advisor who can form a view within weeks rather than months preserves options that would otherwise close.

This is not an argument that internal capability is irrelevant. It is an argument that the structure of a failing programme - its incentives, its politics, its history - often makes internal recovery significantly more difficult. The external advisor's role is not to take over, but to provide the honesty and structure that the programme has lost.

7. KEY TAKEAWAYS

Programme failure is normal, not exceptional. The evidence is consistent and long-standing. Around one in three projects succeed. The rest struggle or fail.

The causes are known and repetitive. Optimism bias, untested assumptions, unmanaged dependencies, governance that confirms rather than challenges, and delivery teams that are unable to escalate bad news.

Failure is usually visible internally before it becomes visible externally. The problem is not detection; it is escalation. Organisations need to make it easier to surface problems, not harder.

Green is often misleading. If the organisation punishes red and treats amber as a problem to be resolved, green becomes the only safe answer. A programme that has been honestly amber for months may be better managed than one that has been green throughout.

Assumptions are hypotheses, not facts. Document them when the case is written, classify them by uncertainty, and set a validation date for each one. Make those dates milestones on the plan, so the sponsor can see whether the assumption has been tested by the point at which it matters. If it has not, that is a decision point, not a surprise.

Dependencies must be actively managed, not tracked. A dependency on a programme you cannot influence is a risk. Be a stakeholder of the programme you

depend on, attend its governance and challenge it about your interests. If that fails, escalate with a recommendation, then invoke the fallback.

Amber becomes permanent when the difficult work is avoided. Risks are documented but not resolved because resolving them requires collaboration with parties outside the programme's control. A programme becomes genuinely green when those conversations have been had, not when the risks have been logged.

Recovery requires diagnosis before action. The first step is not to fix the programme. It is to understand what is actually wrong.

Pre-agreed decisions are powerful. A fallback solution, approved in advance, with a defined trigger, converts a crisis into an agreed change of direction.

External help is often the most effective intervention. Not because insiders are not competent, but because the structure of a failing programme makes it hard for insiders to see problems and say what needs to be done. Advisors who work regularly on failing programmes also recognise the pattern quickly, which matters when the window for recovery is narrowing.

REFERENCES

The Standish Group. *CHAOS Report*. Annual series. Widely cited figures on project success, challenge and failure rates. [The Standish Group](#)

PM World Journal. *Longitudinal analysis of project success rates, 2017 and 2020–2024*. Source for the finding that success rates have stagnated despite the adoption of Agile. [Comparative research on IT project failure rates: A 2025 longitudinal update](#)

McKinsey & Company. *Why do most transformations fail?* Research on transformation failure rates. [Perspectives on transformation | Transformation | McKinsey & Company](#)

National Audit Office. Reports on over-optimism in public sector projects. [Over-optimism in government projects](#)

INSEAD. Research on organisational transformation and defensive organising. [The Perils of Defensive Organising | INSEAD Knowledge](#)

ABOUT AVERRUNCUS

Averruncus is a boutique consultancy led by Daniel Eiloart, a senior transformation and operational finance professional with 20+ years in global financial services. Daniel spent twelve years at HSBC at Director level, including leading the Global Banking & Markets regulatory change portfolio and reporting to the Group Treasurer and Head of Capital Management at Group Treasury, and five years at Accenture as a Senior Manager.

We work in three areas: programme recovery and turnaround, finance transformation and CFO advisory, and digital transformation. Each is grounded in delivery experience rather than theory, and each begins with a practical diagnosis rather than a methodology.

If you would like to discuss a programme that has gone off-track, a finance function that needs to work differently, or a digital change that is not landing, get in touch.

www.averruncus.com