FROM PROBLEM TO VALUE · PART 3 OF 4

TRANSFORMATION RESEARCH PAPER 14 MIN READ

Delivering for value: why on time and on budget is not enough

Many programmes deliver everything they promised and still fail to create value. The fix is to plan, govern and measure delivery around outcomes, not outputs. Part 3 of From Problem to Value, a four-part research series on how organisations create value.



Mayank Paliwal
20 Nov 2026

RESEARCH SERIES · FROM PROBLEM TO VALUE

This is part 3 of 4. 1. Value starts with the problem · 2. The value case · **3. Delivering for value** · 4. Realising value

If you have worked on a large change programme, you have probably seen this. The system goes live on the date promised. The budget is broadly intact. The steering committee thanks the team, the programme closes and its people move on. A year later, the processes it was meant to replace are still running alongside it, the new tool is used by a fraction of the people it was built for, and the benefits in the business case are nowhere to be seen.

By the measures most organisations use, that programme was a success. It still did not create the value it was funded for.

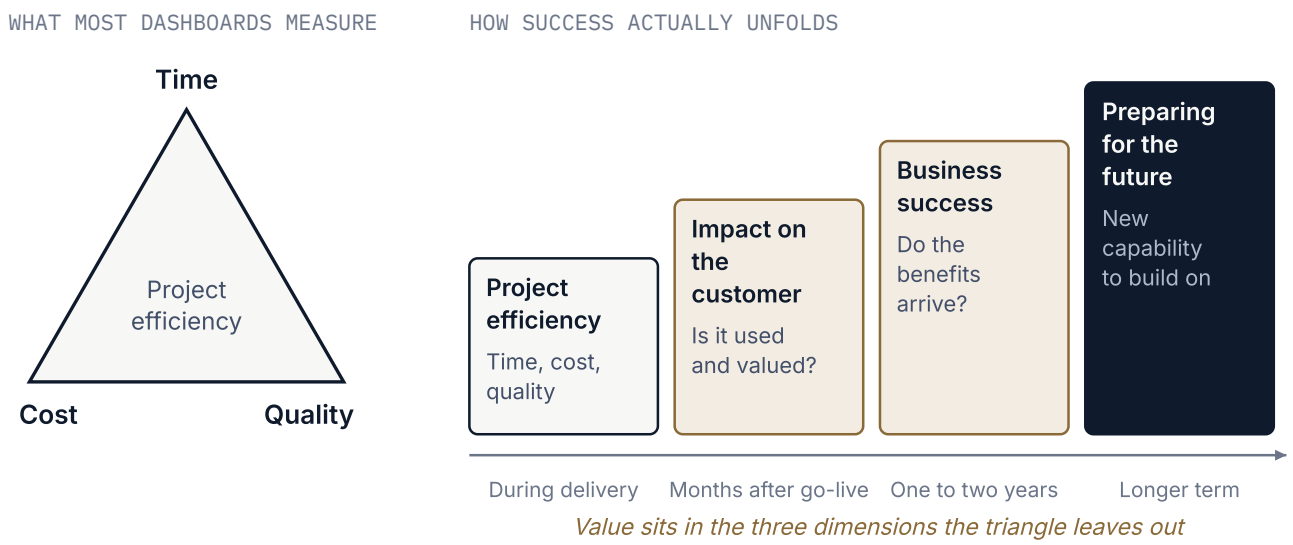
In part 1 I described the value chain of change: value moves from *potential* at the problem stage, to *forecast* in the business case, to *enabled* during delivery, and finally to *realised*. In part 2 I looked at how to build a value case that forecasts honestly. This paper is about the third link, delivery, and the leak that most often opens there: outputs are delivered, but behaviour does not change.

> *A programme that goes live on time with nobody using it has delivered nothing, on schedule.*

The trouble with the iron triangle

For decades, project success was judged against three criteria: time, cost and quality. Atkinson (1999) famously described these as "two best guesses and a phenomenon". Time and cost are estimates made at the point when least is known, and quality is a property that emerges during the work and is hard to define in advance. His wider point was that measuring delivery against the triangle tells you how efficiently a project was run. It says nothing about whether it was worth running.

Shenhar et al. (2001) developed this into a multidimensional view of project success. They argued that success should be judged on at least four dimensions over different timescales: project efficiency in the short term, impact on the customer, business success, and preparing the organisation for the future. The iron triangle only covers the first.



The iron triangle measures how efficiently a project ran. Shenhar et al. (2001) add three further dimensions that play out over months and years, and that is where value is created.

This has a direct consequence for governance. If steering committees and delivery teams are judged on time and cost, they will manage to time and cost. Kerr (1975) described this dynamic in one of the most cited papers in management: organisations routinely reward one behaviour while hoping for another. Reward a programme for going live on the date, and do not be surprised when it protects the date by dropping the training, process redesign and data clean-up that the benefits depend on.

Outputs, outcomes and benefits

The distinction set out in part 1 becomes practical at this stage. A project produces *outputs*: a system, a process, a new organisation design. When people use those outputs in their work, they produce *outcomes*: faster decisions, fewer errors, more satisfied customers. When those outcomes are valued by stakeholders, they become *benefits* (Ward and Daniel, 2012; AXELOS, 2020).

Programme management guidance in the UK has drawn this line for a long time. In *Managing Successful Programmes*, projects deliver outputs and capabilities, while the programme is responsible for transition: embedding those capabilities in the business so that outcomes and benefits follow (AXELOS, 2020). The failure mode is a programme that behaves like a large project, plans to the last output and stops.

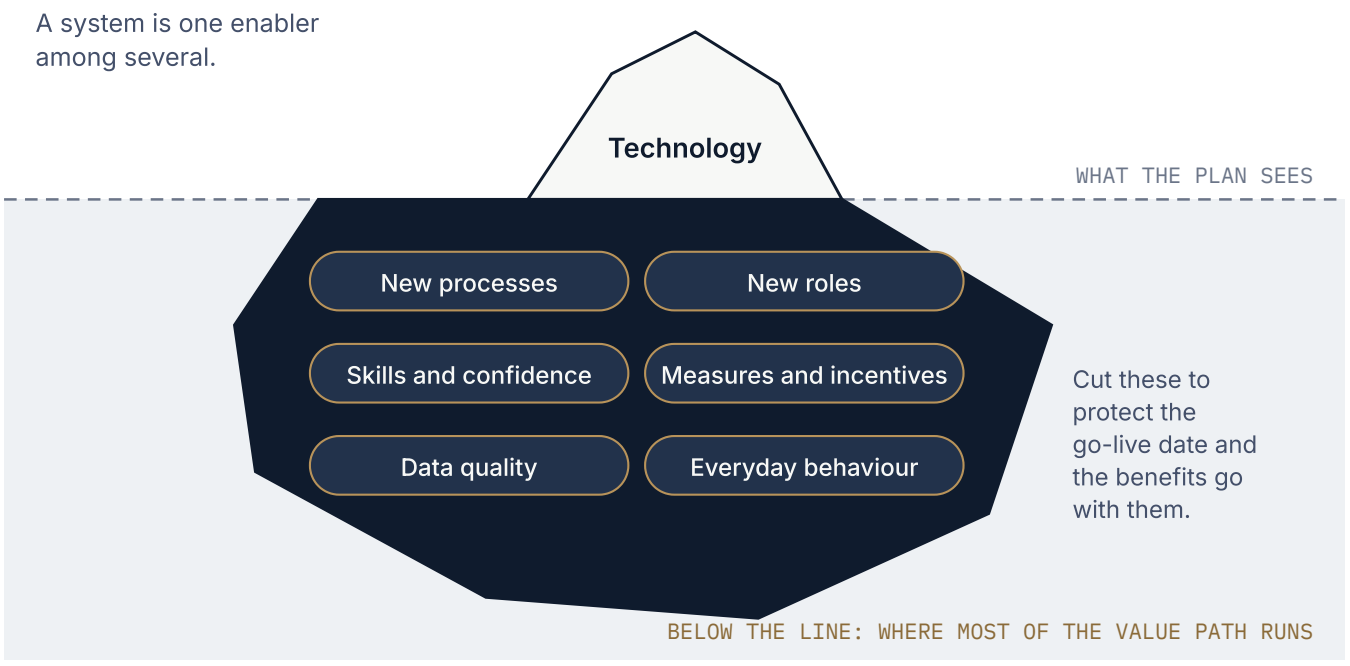
Delivery teams naturally focus on outputs because outputs are what they control. A business analyst can make sure the system meets its requirements. They cannot make a regional sales team use it. That is why delivery for value needs a different design.

Technology is rarely the hard part

Markus (2004) introduced the term *technochange* to describe a particular kind of initiative: one in which IT is used to drive improvements in organisational performance. Technochange is different from an IT project, which aims to deliver a working system, and different from an organisational change programme, which aims to change behaviour without new technology. It combines both, and it fails in a characteristic way. When it is managed as an IT project, the system is delivered but the organisational changes needed to use it well are left to chance.

Research on productivity points the same way. Brynjolfsson, Rock and Syverson (2021) describe a *productivity J-curve* for general-purpose technologies such as AI. Getting value from them requires large complementary investments in intangible assets: new processes, new skills, new business models. These investments are costly and largely invisible in conventional measures, so productivity can appear to fall before it rises. You need the technology for value, but on its own it will not get you there.

The practical lesson is the one the benefits dependency network from part 2 makes visible. Most of the path from investment to benefit runs through business changes. If the delivery plan is organised around the technology, it is organised around the smallest part of the value.



Technochange (Markus, 2004). The technology is the visible part of the work. The complementary changes below the line are what turn an output into an outcome, and they are the first things cut when a plan comes under pressure.

People decide whether value is created

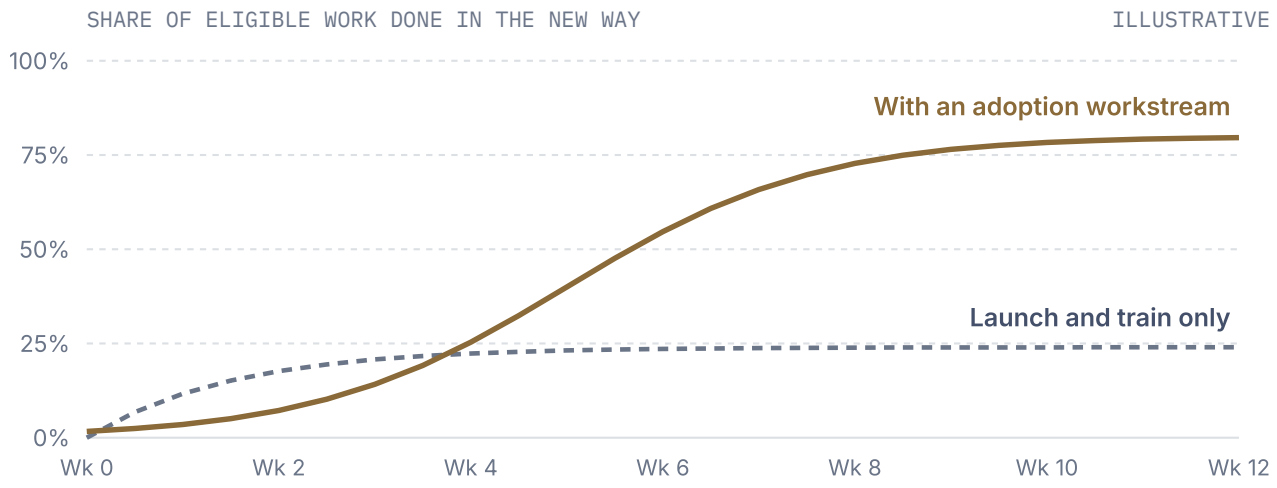
Every benefit in a business case depends, somewhere, on people choosing to work differently. That choice is not automatic, and training is only a small part of it.

Davis (1989), in the work that became the technology acceptance model, found that people's intention to use a new system is driven above all by its *perceived usefulness*, whether they believe it will help them do their job better, and secondarily by its *perceived ease of use*. A system that is easy to use but does not

help is ignored. A system that helps but is hard to use is resented. Either way, the business case suffers.

Change management models break the same idea into stages. Prosci's ADKAR model, for example, holds that individual change requires awareness of the need for change, desire to take part, knowledge of how to change, ability to do so in practice and reinforcement to sustain it (Hiatt, 2006). Most programmes invest heavily in knowledge, through training, and much less in the stages either side of it.

At the organisational level, Kotter (1995) identified a set of common errors in transformation efforts, among them failing to create a sense of urgency, under-communicating the vision and declaring victory too soon. Beer and Nohria (2000) contrasted two theories of change. *Theory E* pursues economic value through hard levers such as structure, systems and incentives. *Theory O* builds organisational capability through culture, learning and engagement. They argued that the most successful companies combine the two instead of choosing one.



The same system, two adoption stories. Training alone builds knowledge. Changing the measures, roles and support around the tool is what moves usage. Figures are illustrative.

For delivery, this means adoption cannot be a phase at the end of the plan. It needs to be a workstream from the start, with its own owner, budget and measures.

Designing delivery around value

These are the practices I have seen make the biggest difference.

Plan backwards from outcomes

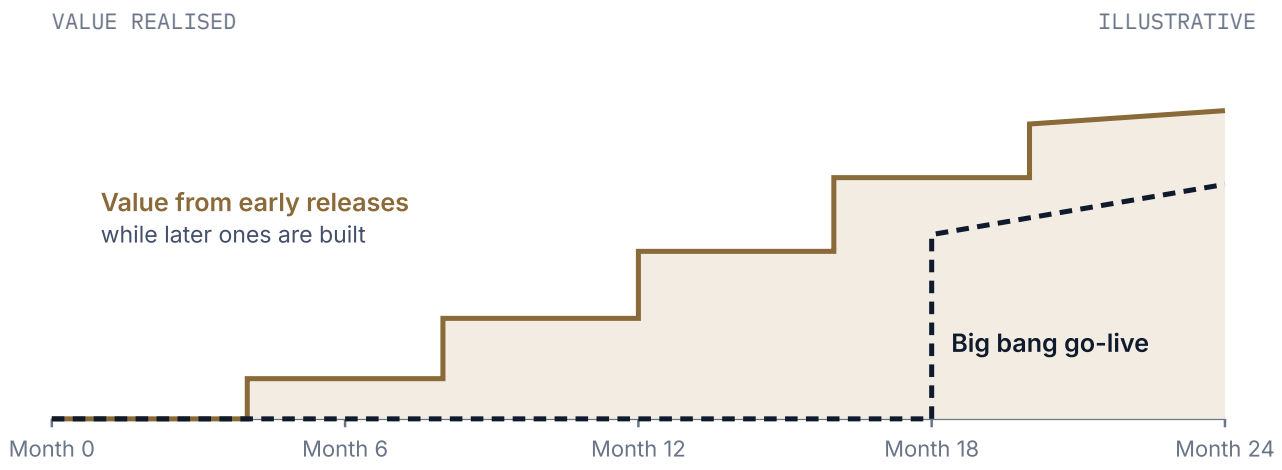
Start the delivery plan from the benefits dependency network instead of the solution architecture. For each benefit, the plan should show the business changes it depends on, who owns them and when they need to be in place. Technology milestones then sit inside that structure as enablers.

Deliver in increments of value

The Agile Manifesto put “working software” ahead of “comprehensive documentation” (Beck et al., 2001). The broader principle is that each release should put something usable in front of real users, so that value starts to flow

and learning starts early. Ries (2011) framed this as a cycle of build, measure and learn, in which each increment is also an experiment.

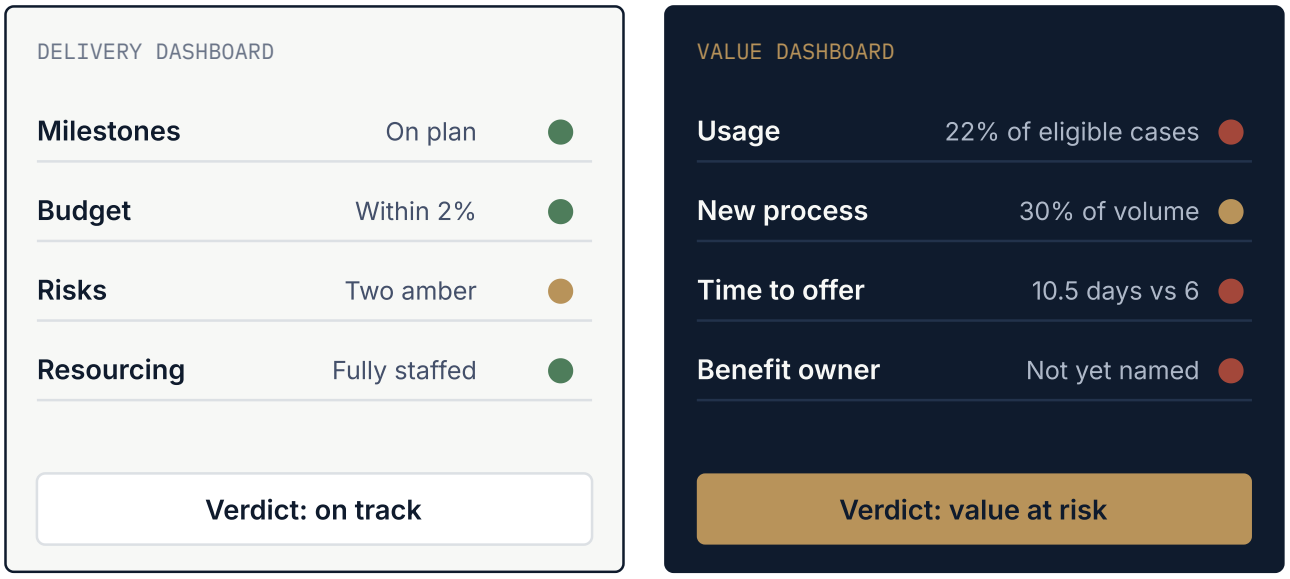
Flyvbjerg and Gardner (2023) reach a similar conclusion from a very different evidence base. Their advice for large projects is to “think slow, act fast”: invest heavily in planning and testing, then deliver quickly in modular steps, because the longer a project is exposed to the world, the more can go wrong. Modularity also means value can be realised from early modules while later ones are still being built.



Delivering in increments of value. The shaded area is value the organisation gets that a single go-live would have delayed, and every release is also a chance to learn. Figures are illustrative.

Put outcome measures on the dashboard from day one

If the only indicators on a programme dashboard are milestones, budget and risks, the programme will be managed as a project. Add two or three leading indicators of value from the start: usage, the share of work flowing through the new process, early movement in the target outcome. These should be measured against the baseline set in the value case.



The same programme, the same week. A dashboard that only tracks delivery says all is well. Adding a handful of value measures, each against a baseline, tells the steering committee where to act. Figures are illustrative.

Govern for decisions

A steering committee that sees outcome measures can make value decisions: to delay go-live until the process is ready, to fund extra adoption support, or to stop a workstream that is not moving the numbers. One that sees only status reports can only approve or worry. I have written about this in Good programme governance makes decisions faster and Most steering committees watch. Few of them steer. The SteerCo Pack template is designed around decisions instead of updates.

Make it safe to report bad news

Outcome measures only help if people are willing to report them honestly. Edmondson (1999) showed that teams with higher *psychological safety*, a shared belief that it is safe to take interpersonal risks, engage more in learning behaviours such as asking for help, admitting mistakes and raising concerns. Programmes with a culture of reporting green until the last moment lose the

chance to fix adoption problems while they are still cheap. Leaders set this culture through how they respond the first time someone reports a problem.

Protecting the value during delivery

The most dangerous moment for value in delivery is when the plan comes under pressure. Something has to give, and the easiest things to cut are the ones that do not appear on the critical path for go-live: process redesign, training, data clean-up, communications, the change network.

Those are the activities that turn outputs into outcomes. A useful rule is that any change to scope, timing or budget is assessed against the value case as well as the plan. The question becomes: if we cut this, which benefit is at risk, by how much, and who has agreed to carry that risk? A RAID log that records benefit impact alongside delivery impact makes this visible.

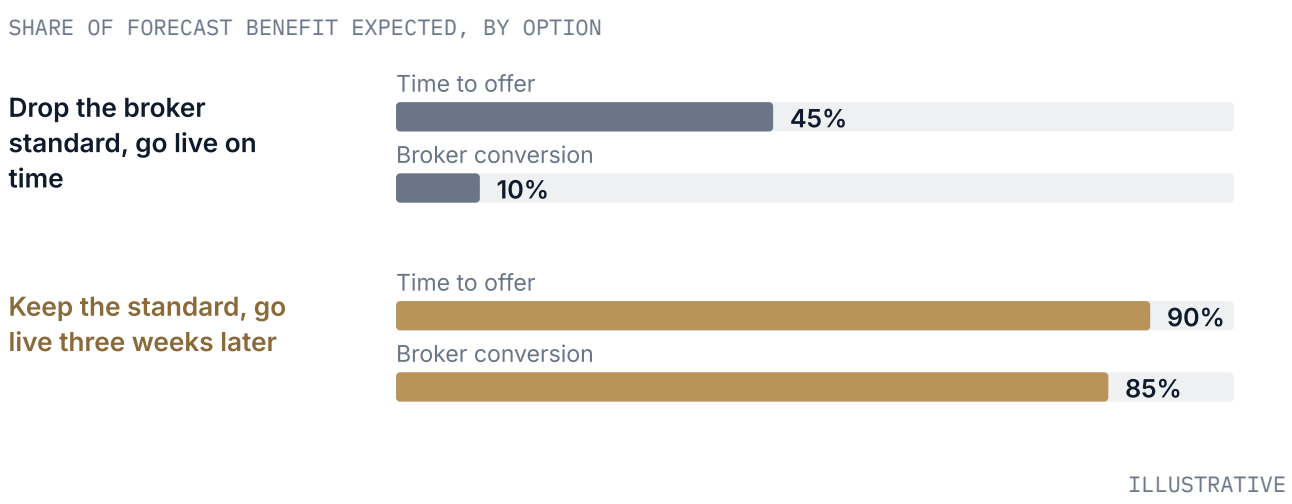
A worked example

Return to the mid-sized UK lender from parts 1 and 2. The board funded a targeted option to cut mortgage time to offer: automated document checks, workflow changes in underwriting and a new standard for communicating with brokers.

Three months into delivery, the programme came under time pressure. The technology team proposed dropping the broker communication standard from the first release so that the document checks could go live on schedule. On the milestone dashboard, this looked sensible. The standard was a “soft” deliverable, and the technology was on track.

The value dashboard told a different story. The benefits map from the business case showed that broker conversion, the largest single benefit, depended on

brokers receiving clearer requests for missing documents. Without the new standard, the document checks would flag gaps faster, but brokers would still be chased in the old, confusing way. Time to offer would improve a little. Conversion would barely move.



What the value dashboard showed the steering committee. Protecting the date would have kept the milestone green and lost most of the largest benefit. Figures are illustrative.

The steering committee kept the standard in scope, delayed go-live by three weeks and assigned a named owner in the broker team. When the release went live, usage of the new document checks reached most eligible cases within two months, and time to offer in the pilot region fell markedly. The three-week delay was a cost. Dropping the standard would have cost most of the benefit.

The Change Impact and Stakeholder Map and Change Communications Plan templates help to plan this kind of business change alongside the build.

TRY THIS

Look at the dashboard for a programme you sponsor or run. Count the measures that tell you about delivery (milestones, budget, risks) and the measures that tell you about value (usage, process adoption, movement in the target outcome). If the second number is zero, add two value measures before the next steering committee, each against a baseline.

What this means for the rest of the chain

Delivery for value is where *forecast* value becomes *enabled* value. Done well, it hands the business a capability that people are already using, with outcome measures starting to move and owners in place. Done badly, it hands over a system and a closure report, and leaves the business to find the benefits on its own.

Even a programme that does all of this well has not yet created value. Enabled value is still potential. It is realised only when the improvement shows up in the numbers the organisation cares about, and it is sustained only if someone keeps it there. In part 4, Realising value, I turn to the final link in the chain: who owns the benefits after the project team has gone, and how to make sure they arrive and stay.

References

Atkinson, R. (1999) ‘Project management: cost, time and quality, two best guesses and a phenomenon, its time to accept other success criteria’, *International Journal of Project Management*, 17(6), pp. 337–342.

AXELOS (2020) *Managing Successful Programmes*. 5th edn. London: TSO.

Beck, K. et al. (2001) *Manifesto for Agile Software Development*. Available at: https://agilemanifesto.org (Accessed: 20 November 2026).

Beer, M. and Nohria, N. (2000) 'Cracking the code of change', *Harvard Business Review*, 78(3), pp. 133–141.

Brynjolfsson, E., Rock, D. and Syverson, C. (2021) 'The productivity J-curve: how intangibles complement general purpose technologies', *American Economic Journal: Macroeconomics*, 13(1), pp. 333–372.

Davis, F.D. (1989) 'Perceived usefulness, perceived ease of use, and user acceptance of information technology', *MIS Quarterly*, 13(3), pp. 319–340.

Edmondson, A. (1999) 'Psychological safety and learning behavior in work teams', *Administrative Science Quarterly*, 44(2), pp. 350–383.

Flyvbjerg, B. and Gardner, D. (2023) *How Big Things Get Done*. London: Macmillan.

Hiatt, J.M. (2006) *ADKAR: A Model for Change in Business, Government and Our Community*. Loveland, CO: Prosci Learning Center Publications.

Kerr, S. (1975) 'On the folly of rewarding A, while hoping for B', *Academy of Management Journal*, 18(4), pp. 769–783.

Kotter, J.P. (1995) 'Leading change: why transformation efforts fail', *Harvard Business Review*, 73(2), pp. 59–67.

Markus, M.L. (2004) 'Technochange management: using IT to drive organizational change', *Journal of Information Technology*, 19(1), pp. 4–20.

Ries, E. (2011) *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. New York: Crown Business.

Shenhar, A.J., Dvir, D., Levy, O. and Maltz, A.C. (2001) ‘Project success: a multidimensional strategic concept’, *Long Range Planning*, 34(6), pp. 699–725.

Ward, J. and Daniel, E. (2012) *Benefits Management: How to Increase the Business Value of Your IT Projects*. 2nd edn. Chichester: John Wiley & Sons.



Mayank Paliwal

Founder, InflexionIQ

Mayank works on business transformation, strategy and operating model design. He is an independent consultant and founder of InflexionIQ, previously at Deloitte, with a master's in Strategy & Innovation from Saïd Business School, University of Oxford.