

BEYOND HEADCOUNT

Resource Allocation in Data Transformation: Why Headcount Is Not the Same as Capacity

A Practitioner's Perspective on Capacity and Critical Skills in Data Transformation

Giovanni Leonardi

March 2019



Resource Allocation in Data Transformation: Why Headcount Is Not the Same as Capacity

Portfolio · No. 88

Originally written March 2019 | Re-edited and re-rendered for the WhereBeyond Edition, 2025

*“Headcount tells you how many people you are paying.
Capacity tells you how fast you can go. Only one of
them is on the status report, and it is the wrong one.”*

1. The Plan Says Fully Resourced. The Programme Says Otherwise.

There is a particular kind of programme status that ought to worry anyone who has run a large data transformation. The resourcing line is green. Every workstream shows its full complement of people. Headcount is at plan, the roles are filled, the org chart has no gaps. And yet the programme is not moving. Milestones slip, dependencies pile up, and the same few decisions and designs sit unresolved week after week. On paper it is fully resourced. In reality it has almost no spare capacity where capacity actually matters.

I have watched this happen often enough to be confident it is not bad luck. It is the predictable result of confusing two things that a resourcing report treats as identical and that a delivery reality treats as entirely different: headcount and capacity. Headcount is how many people are assigned. Capacity is how much of the work that actually gates delivery can be done at once. In a data programme these two numbers are rarely the same, and the gap between them is where transformations quietly stall.

2. Where The Real Constraint Lives

The reason the two diverge is that data transformation does not consume generic effort. It consumes specific, scarce skills, and it consumes them at particular points. A programme can be swimming in capable people and still be throttled by a handful of roles that everything depends on.

- Architects. The design decisions that unblock everyone downstream sit with a small number of people who understand the whole estate. If there are two of them and forty things need their judgement, forty things queue behind two people, regardless of how many engineers are waiting to build.
- Senior engineers. Not all engineering capacity is equal. The hard integration, the migration logic, the performance problem – these fall to a few, and the rest of the team is often blocked waiting on their output rather than working in parallel with it.

- Business subject-matter experts. The people who actually know how the data is used, what a field means, and which exceptions matter are usually doing their day jobs. They are lent to the programme part-time, and their scarce, fragmented attention becomes a bottleneck no amount of contractor headcount can relieve.
- Decision-makers. The scarcest resource of all is often the authority to decide. Scope, priority, trade-off and sign-off decisions concentrate on a few executives whose availability, not the team's effort, sets the true pace of the programme.

Every one of these is a role where you cannot solve a shortage by adding bodies elsewhere. Ten more testers do not help if the work is queued behind one architect. That is the essence of the problem: capacity is set by the scarce, critical skills, and headcount is measured across all skills as though they were interchangeable. They are not.

Headcount is measured across all roles as if they were interchangeable. Capacity is set by the two or three roles that everything depends on. The distance between those two facts is where fully-resourced programmes stall.

3. A Concrete Illustration

Consider a workstream that looks, on the resourcing report, entirely healthy.

Role	Assigned (headcount)	Genuinely available to the critical path
Delivery lead	1	1
Engineers	8	8
Test analysts	4	4
Solution architect	1	0.3 (split across three workstreams)
Business SME	2	0.5 (part-time, day job first)
Decision-maker / sponsor	1	0.2 (available for one forum a fortnight)

The report shows seventeen people assigned and a full team. The delivery reality is that the entire workstream advances at the speed of three-tenths of an architect, half a subject-matter expert, and a sponsor who can take a decision once a fortnight. The eight engineers and four testers are not the constraint; much of the time they are idle against the critical path, or busy on work that will need rework once the scarce roles catch up. The workstream is not under-resourced in headcount. It is starved of capacity in precisely the three roles that gate it.

This is why a programme can report green on resourcing and red on delivery at the same time, and why adding more of the plentiful roles - which is what a headcount view invites you to do - makes the picture worse, not better. More engineers waiting on one architect is more cost, more coordination, and more rework, for no more throughput.

4. How To See The Constraint Before It Sees You

The discipline that separates programmes that manage this from programmes that are managed by it is the willingness to look past the headcount line to the capacity beneath it. In my experience three habits do most of the work.

1. Map capacity by critical skill, not by total headcount. For each workstream, identify the two or three roles that everything depends on, and measure their genuinely available time on this programme - net of split assignments, day jobs and leave. That number, not the total, is the workstream's real capacity. Track it as the metric that matters.
2. Find the bottleneck and sequence around it. Once the scarce roles are visible, the constraint reveals itself: it is whatever the most work is queued behind. Sequence deliberately so that scarce capacity is spent on what unblocks the most downstream work first, and do not start work that will simply pile up behind a constraint that cannot clear it.
3. Protect and multiply the scarce roles, do not dilute them. The instinct to keep a scarce architect in every meeting or across five workstreams destroys the very capacity that is constraining you. Protect their time for the decisions and designs only they can do, capture and reuse those decisions so they are not asked twice, and grow the next architect deliberately - because the only durable way to add capacity in a scarce role is to create more of it.

“Ten more testers do not help if the work is queued behind one architect. You do not relieve a bottleneck by adding capacity somewhere else.”

5. Why The Headcount View Persists

If the distinction is this clear, it is worth asking why the headcount view is so stubborn. Part of the answer is that headcount is easy to count and capacity is not. A resourcing report can be assembled from the org chart in an afternoon; a true capacity view requires someone to know which roles gate which work and how much of each scarce person is genuinely available. The easy number wins because it is available, and the available number quietly becomes the number that is believed.

Part of the answer is financial. Budgets are built and defended in headcount, and a fully deployed headcount reads as good stewardship. A leadership team that has funded a hundred roles wants to see a hundred roles working, and a report that shows them all assigned is more comfortable than one that admits three of them are the only ones that matter this quarter. The headcount view flatters the budget, and the capacity view complicates it.

And part of the answer is simply that the failure is invisible until late. A programme starved of capacity in its scarce roles does not fail loudly at the start. It looks fine, then it looks slightly behind, then it is a year late with no single moment anyone can point to. Because every resourcing report along the way was green, the diagnosis arrives too late to be cheap. The teams that avoid this are the ones that refuse to be reassured by a full org chart and insist on seeing the capacity underneath it from the first month.

6. The Uncomfortable Implication For Portfolio Planning

There is a portfolio-level lesson in this that is easy to resist. If genuine capacity is set by a handful of scarce roles, then the number of data initiatives an organisation can actually run at once is far smaller than its total headcount suggests. A portfolio planned on headcount will always start too many things, because the plentiful roles look available. Each new initiative then draws on the same scarce architects, senior engineers, subject-matter experts and decision-makers, and the whole portfolio slows together while every individual resourcing report stays reassuringly green.

The organisations that get more done are usually the ones running fewer things at once, sequenced around their genuinely scarce skills, rather than the ones with the largest programmes. That is a hard message to deliver to a leadership team that has budgeted for headcount and wants to see it all deployed. But it is the difference between a portfolio that is busy and a portfolio that delivers.

The correction is not more people. It is the honesty to plan around the constraint you actually have. Count capacity where it is scarce, sequence the work to spend that capacity well, and resist the temptation to read a full org chart as a fast programme. Headcount tells you how many people you are paying. Capacity tells you how fast you can go. Only one of them is on the status report, and it is the wrong one.