The Work Moved to India. The Relationships Didn't.

What onboarding can't transfer, and why AI will make it harder for GCC leaders.

RESHMI NANDY · reshminandy.com · October 2026

A business leader returns from a technology symposium with an idea. He doesn't book time with the new capability lead in India. He calls the engineer he has worked with for ten years, in their shared language, over coffee.

That moment is what most GCC transitions are not designed for. When an enterprise moves critical IT capability to India, it usually keeps a core of its best people: those who hold the business knowledge, the history and the relationships. A structured onboarding and integration KT moves their tasks. It cannot move their standing with the business.

I learned this when I took over a design-and-build capability from India, leading a team of exactly such people across Europe and South America.

What onboarding does well

To be fair, leadership onboarding and integration KT have never been better. The programme I joined was generous: visits to regional sites and factories, sessions with IT leads and some business leads, the history of the enterprise. Much of that history is now a search away; AI can assemble an enterprise's past, its peers and its press coverage in an afternoon. Senior colleagues can tell you which crises happened and who resolved them. I used that time to build an inventory and knowledge repository of more than 450 integration interfaces, and then began modernising them. Systems, documents and history transfer well. What they sit inside does not.

What it can't transfer

Shared language and easy conversation. A new member from another culture makes early conversations more careful. Business leaders wonder whether an inside joke will land, or whether a familiar stereotype will sound politically incorrect. It seems trivial. In the early brainstorming phase, when ideas are half-formed and people think aloud, it decides who gets invited.

Informal channels. Business leaders and long-standing team members often know people outside the company from earlier projects. They meet offline and brainstorm, and vendor representatives join with their own agenda. Some decisions are effectively made there. When I joined, the platform for a new engagement layer had already been chosen: a proprietary low-code platform for which India had almost no talent at the time. I had to staff it with local hires onshore, and the India skills base never caught up during my tenure. The decision quietly worked against the very GCC model it was meant to serve. The real issue, I understood later, was governance: wherever business leaders chose technology without the enterprise architecture function at the table, the same pattern appeared.

The vendor rift. Most enterprises run a vendor mix: well-liked specialist firms that take a small share of the budget for digital design, ERP or industry platforms, and large IT services providers that run infrastructure and applications and take most of it. A GCC move shifts the centre of decision-making, and the large providers feel it first. Having since worked on the vendor side, I understand why. An account nurtured for years by an onshore team does not hand its revenue to an India team gracefully, even when the customer wants the work in India. Until the provider settles that internally, its onshore team lobbies the client's onshore leaders and tries to run the account as it always has. Knowledge transfer slows; key people become hard to reach. It is rational behaviour, not bad faith, and the new capability lead still needs that vendor, because in most large enterprises the tribal knowledge sits with vendors, not employees. Emotional intelligence helps. The real lever is commercial.

Legitimacy. Even when a newly hired India-based engineer or analyst was clearly capable of cross-functional decisions, with empathy and commercial sense, they were kept out of early discussions for almost nine months. Several came to me with concerns: their stand-ups felt ambiguous, with a lot of assertion about who held power. I saw the pattern within three months, but fixing it meant getting every stakeholder to change how they communicated, and that never happens on a given date. My sponsor would coach people, but he would not force a new process on business leaders. In most enterprises, business satisfaction outranks process, and cost becomes a priority only when the CFO makes it one.

Trust built in a crisis. That kind of trust can't be scheduled. It has to be earned, and the opportunity arrives without warning.

The trap

The trap for an India-based capability lead is to become the allocator: the person who assigns work while the real conversations flow around her. Enthusiastic team members started prototyping ideas from those coffee conversations without

telling me. Their attention drifted from committed work to discovery. Asking directly ("Are you working on something I don't know about?") rarely helps before functional trust exists.

An organisational change made it harder. The role I took on had previously been an allocator's role; the enterprise had redefined it to need an enterprise architect leading solution architects across technology streams. The org chart changed; the business's habits didn't. Leaders kept engaging the role the old way, and long-standing team members often knew where the business was heading before I did.

So I built a listening post. I started a fortnightly call with our most influential vendor partners and my team, to hear what they were discussing and how they were positioning new technology. It surfaced much of the informal information I was missing. I reminded myself that every business leader is right to want a great product. My job was to learn early, bring in the stakeholders the decision would affect (above all the run leaders), and use a quick run-cost assessment to decide whether the direction was viable or whether I should step in.

What worked

Three things changed the dynamic. First, candour upward. I described the problem plainly to my supervisor, who had strong relationships across the company, and he coached business leaders and senior team members. Not every leader gets that support.

Second, a crisis. A major incident escalated past the run team and landed with me. A daily trading P&L, built overnight in US dollars from an American price feed and due before the next morning, was failing. The feed arrived and was stored, then a forgotten procedure, triggered through an old workaround, deleted the data before the P&L could run: pure, unidentified technical debt. I stayed on a call with the global team for 24 hours, from the office and then from home in India, until we found and fixed it. The next batch ran about 40 hours later without the problem returning. People noticed who stayed.

Third, co-location. Over time I sent five people to work onshore for at least three months each, two at a time: three India-based team members and two critical vendor engineers. The aim was for business leaders to get used to discussing technology and ideas with them. At first, leaders still contacted onshore team members directly and reached India-based members through me. Then the feedback started coming to us directly: business leaders said the India team was highly effective. That was the point at which the India tech lead I had hired was confirmed to lead our Agile competency, and the capability stopped depending on me.

What I would build into any GCC transition

If I were designing a GCC transition today, I would add six things the formal onboarding plan leaves out:

1. **Vendor conversations:** who is pitching what to senior leaders, and what they are hearing.
2. **The transformation vision and its constraints,** discussed with business leaders, not inferred from documents.
3. **The failed initiatives.** They never appear in a KT plan, and they explain most of today's resistance.
4. **A stakeholder map by actual influence, not role.** Who owns a decision and who shapes it are often different people, and a celebrated new hire from a top consultancy or a competitor can redraw that map overnight.
5. **Explicit decision rights and a co-location budget** for the India-based lead, treated as transition costs rather than perks.
6. **Vendor transition terms:** knowledge-transfer acceptance criteria, key-personnel commitments and milestones in the contract, with knowledge transfer tested by outcomes (can the new team resolve real tickets unaided?) rather than by documents delivered.

Why AI makes this harder

AI raises the stakes on every one of these. Business users now prompt AI tools directly, so early ideas may never reach the capability team at all. AI-augmented stars need a leader even less. And the hardest question in enterprise AI is not technical: it is agreeing what success looks like. Accuracy metrics are necessary, but they are not what a business leader calls success.

An India-based capability lead without trust is exposed here. If AI investments don't show a return quickly, the "could not deliver ROI" label sticks fastest to the leader furthest away, and the remote capability lead is the easiest to replace. Another enterprise's AI success story won't help. AI success is new, standards are still forming, and one enterprise's win does not transfer to another by default.

What helps is a success contract agreed with the business before anything is built: the outcome to move (cycle time, cost to serve, errors reaching customers, how often people override the AI), the baseline, and when it will be measured. Then come small, visible wins, and a grind that every stakeholder lives through together.

Then there is the knowledge itself. In most large enterprises it sits with vendors and veterans, and documentation SLAs have rarely captured it; even where they exist, quality is debatable. AI changes the economics. Turning documents, ticket history and meeting records into an enterprise knowledge context, a graph of events, decisions and people, becomes one of a new leader's first jobs. It needs an owner, and it should be tested by what the team can actually do with it. Done well, a new leader starts with the network map instead of discovering it one coffee at a time.

I fully expect my team members to know more than I do in their own areas. My job is to connect what they know to where the enterprise is going, and to keep everyone's eyes on the future. In a GCC, that only works once the relationships have moved along with the work.

Reshmi Nandy is an enterprise transformation, AI and GCC leader based in Bengaluru. More writing at reshminandy.com.