



BEYOND THE ORG CHART

Platform Teams Versus Project Teams — The Structural Tension

A Practitioner's Perspective on Organisational Design for Cloud in the Enterprise

Giovanni Leonardi

July 2016

Platform Teams Versus Project Teams — The Structural Tension

Programme · No. 164

Originally written July 2016 | Re-edited and re-rendered for the WhereBeyond Edition, 2025

“The project team is optimised to finish; the platform team is optimised to endure — and most organisations have no mechanism for resolving the tension between these two entirely legitimate objectives.”

1. The Two Models Nobody Chose

Somewhere in the last eighteen months, most large organisations acquired two kinds of technology team without ever deciding to create either of them. The first is the platform team — a small, persistent group that builds and operates a shared cloud environment, maintains the tooling, defines the standards, and keeps the lights on. The second is the project team — a temporary, cross-functional group assembled to deliver a specific business outcome, funded for a defined period, and disbanded when the outcome is achieved or the budget runs out.

Both models are rational. Both have deep roots in how organisations have structured technology work for decades. And in the context of cloud adoption, they are in fundamental tension with each other — a tension that most organisations have not yet recognised, let alone resolved.

The platform team wants stability, consistency, and long-term thinking. It wants to build once and reuse many times. It measures success in uptime, adoption rates, and the declining cost of provisioning new environments. The project team wants speed, autonomy, and the freedom to make technology choices that serve its immediate objectives. It measures success in features delivered, deadlines met, and business outcomes realised.

Neither is wrong. But the structural tension between them is real, and it is shaping cloud adoption in ways that deserve more attention than they are receiving.

2. Where the Platform Team Comes From

The platform team is not a new idea. It is the latest incarnation of a pattern that has recurred across every major infrastructure shift: the shared services team, the middleware team, the enterprise integration team. Each time a technology becomes

foundational enough that multiple project teams need it, someone creates a team to provide it as a service.

In the cloud context, the platform team typically emerges from one of two sources. In some organisations, it grows out of the infrastructure team that managed the data centre. These are the people who understand networking, storage, compute, and the operational disciplines that keep infrastructure running. They see cloud as a new delivery mechanism for the same fundamental capabilities, and they set about building a governed, standardised cloud environment that project teams can consume without having to understand the underlying complexity.

In other organisations, the platform team emerges from a successful project team. A group that built something well on cloud — usually the first significant cloud workload — gets asked to help others do the same. Gradually, their role shifts from building applications to building the platform that other applications run on. They bring a developer's sensibility to infrastructure: automation over manual process, code over configuration documents, iteration over big design up front.

These two origins produce very different platform teams, with different instincts, different tools, and different relationships with the project teams they serve. The infrastructure-origin platform team tends toward control and standardisation. The project-origin platform team tends toward enablement and flexibility. Neither instinct is wrong, but the difference matters enormously for how the organisation experiences cloud.

3. Where the Project Team Collides

The project team, meanwhile, is under pressures that have nothing to do with platforms. It has a sponsor who wants results. It has a deadline that was set before the technical approach was understood. It has a budget that assumes a linear relationship between time and delivery. And it has just discovered that the platform team's environment does not support the specific cloud service it needs for a critical feature.

This is the moment where the structural tension becomes visible. The project team sees the platform as a constraint. The platform team sees the project's request as a threat to the stability and security of a shared environment. Both are correct.

The pattern plays out with remarkable consistency. The project team requests a new cloud service — perhaps a managed database service, or a container orchestration capability, or a specific machine learning API. The platform team evaluates the request against its standards, its security requirements, and its operational capacity. The evaluation takes time. The project team cannot wait. One of three things happens.

In the first scenario, the project team waits, the timeline slips, and the sponsor escalates. The platform team is blamed for being slow, bureaucratic, and out of touch with business needs. In the second scenario, the project team routes around the platform — provisioning its own cloud account, outside the governed environment, using a corporate credit card. The feature ships on time, but the organisation now has an ungoverned cloud workload that nobody in the platform team knows about. In the third scenario, the platform team fast-tracks the request, adding the service to the

platform without its usual due diligence. The feature ships on time, but the platform now has a component that has not been properly assessed for security, cost, or operational supportability.

None of these outcomes is good. All three are happening, simultaneously, in most large organisations I observe.

4. The Funding Model Makes It Worse

Underneath the structural tension is a funding problem that most organisations have not confronted. Project teams are funded by business cases — defined investments with expected returns, approved through capital expenditure processes that have been refined over decades. Platform teams are funded as operational overhead — a cost to be minimised rather than an investment to be optimised.

This asymmetry has consequences. The project team has a mandate to spend money to deliver an outcome. The platform team has a mandate to keep costs down while serving an expanding set of consumers. As cloud adoption accelerates — and it is accelerating faster than most organisations anticipated — the platform team's workload grows, but its funding does not grow proportionally. The result is either declining service quality, or a platform team that becomes a bottleneck not because it is slow but because it is overwhelmed.

The funding model also creates a perverse incentive around cloud services. Every new service the platform team adds to its supported catalogue increases its operational burden. But project teams are constantly requesting new services because the cloud providers are releasing them at a pace that makes last year's catalogue look incomplete. The platform team is caught between the pressure to keep up and the reality that every addition has an ongoing cost.

In my experience, the organisations that manage this tension well are the ones that have moved the platform team from an operational cost model to a product model — treating the platform as an internal product with a roadmap, a backlog, and investment funding tied to the value it enables rather than the cost it incurs. But this shift requires a change in how technology leadership thinks about shared capabilities, and that change is happening slowly.

5. What the Tension Reveals

The structural tension between platform teams and project teams is not, at its root, a cloud problem. It is an organisational design problem that cloud has made visible and urgent.

It reveals that most organisations have not updated their team structures, funding models, or governance processes to reflect the reality of cloud-era technology delivery. They are running a new operating model on old organisational machinery, and the friction is showing.

It reveals that the distinction between "building things" and "running things" — the distinction that underpinned the project/operations split for twenty years — is breaking down. In a cloud world, building and running are not sequential phases; they are concurrent activities performed by overlapping teams. The project team builds the application but relies on the platform team to run the infrastructure. The platform team builds the platform but relies on project teams to validate that it meets real needs. The handoff model that worked when infrastructure was physical does not work when infrastructure is code.

"The project team is optimised to finish; the platform team is optimised to endure — and most organisations have no mechanism for resolving the tension between these two entirely legitimate objectives."

It reveals, perhaps most importantly, that cloud adoption is an organisational transformation, not a technology migration. The technology is the easy part — provisioning virtual machines on AWS or Azure is not meaningfully harder than provisioning them in a data centre. The hard part is redesigning the human systems — the teams, the funding, the governance, the career paths, the incentives — to work with cloud rather than against it.

6. The Patterns That Are Emerging

Across the organisations where I have observed this tension playing out, a few patterns are beginning to emerge that suggest a direction, if not yet a solution.

The platform-as-product pattern. The most effective platform teams are the ones that think of themselves as product teams. They have a product owner who prioritises the backlog based on consumer demand. They have a roadmap that is informed by what project teams need, not just by what the platform team thinks is architecturally correct. They measure adoption, not compliance. This is a significant cultural shift for teams with an infrastructure heritage, but the organisations that have made it report materially better relationships between platform and project teams.

The embedded engineer pattern. Some organisations are experimenting with embedding a platform engineer in each major project team. This person acts as a bridge — translating the project team's needs into platform requests, and explaining the platform's constraints in terms the project team can work with. The model is expensive in headcount terms, but it dramatically reduces the friction at the interface.

The self-service with guardrails pattern. Rather than requiring project teams to request services through the platform team, some organisations are building self-service catalogues with automated guardrails. The project team can provision any approved

service directly, and the guardrails enforce security, cost, and compliance standards automatically. This is the most promising pattern, but the tooling to support it is still immature in 2016, and the organisational trust required to let project teams self-serve is hard won.

The two-speed compromise. Some organisations have accepted the tension rather than trying to resolve it. They run a governed platform for production workloads and a lightly governed sandbox for development and experimentation. Project teams can move fast in the sandbox and are subject to platform governance only when they promote to production. This is pragmatic, but it creates its own problems: workloads built in the sandbox may not be compatible with the production platform, and the promotion process becomes a new bottleneck.

7. Why This Will Not Resolve Itself

The structural tension between platform teams and project teams is not a temporary friction that will smooth itself out as organisations mature in their cloud adoption. It is a permanent feature of how enterprises deliver technology in the cloud era, because it reflects a genuine trade-off between two things organisations need: the agility to deliver business outcomes quickly, and the stability to operate technology reliably at scale.

The organisations that will manage this tension most effectively are the ones that name it, acknowledge it, and design their structures, funding, and governance explicitly around it. The ones that pretend it does not exist — or that treat it as a problem to be solved by choosing the right methodology — will find that it expresses itself in shadow cloud adoption, in escalations, in frustrated project sponsors, and in platform teams that burn out.

“Cloud did not create the tension between those who build and those who sustain. It removed the buffers — the long procurement cycles, the physical separation of development and operations — that used to keep the tension hidden.”

The question is not whether an organisation should have platform teams or project teams. It will have both. The question is whether it designs the relationship between them deliberately, or discovers it through a series of painful collisions. In my experience, most organisations are still in the collision phase. The design phase is the work ahead.