

March 2021

Enterprise Infrastructure is going Agile

But we have to drop the Plan-Build-Run model

Contents

- 03** Introduction
- 04** What is the role of the enterprise infrastructure function in a Cloud 1st organization?
- 06** What is the Plan-Build-Run ('PBR') model and why is it no longer optimal?
- 08** What is the Customer-Centric Model?
- 11** Are you on the right track?
- 12** About Citihub Digital

Author

Chris Allison
chris.allison@citihub.com

Introduction



Infrastructure operating models have typically followed the Plan-Build-Run model (otherwise known as ‘Plan-Build-Operate’) over the last two decades but it is synonymous with ‘Waterfall’ processes, and friction has been increasing with business-aligned application teams as they have embraced Agile. Forward-thinking firms are now re-organizing their infrastructure teams around end-to-end product ownership in a ‘Customer-Centric’ model as they start to embrace Agile. But many leaders are finding it difficult to let go of the stability and control that the Plan-Build-Run model has provided.

Technology infrastructure departments for large corporates typically have thousands of staff, especially the larger banks we deal with so finding an efficient model can have a direct impact on share price. Getting the operating model right can also be the difference between accelerating or stalling your firm’s digital transformation.

What is the role of the enterprise infrastructure function in a Cloud 1st organization?



We sometimes hear the argument that infrastructure groups should largely disappear once apps are migrated to the cloud. Within this argument, firms would be fully reliant on externally hosted services and key infrastructure engineers would be folded into SRE roles as part of agile squads alongside developers and business representatives.

We understand the sentiment and the desire to disintermediate a group that can be seen as a source of friction, particularly if you're looking through the lens of an Agile team. But these infrastructure groups should exist well into the next decade and continue to be an enabler of digital transformation alongside their role managing legacy debt although they may split into two primary parts as described below. At the very least, they need to provide the end user technologies and networks needed to access business and corporate apps. Beyond that, a central function still needs to stitch together automation and controls to safely access a diverse portfolio of services from multiple cloud providers.

Infrastructure groups today are large for two main reasons, both of which will be eroded in the cloud model:

1. They support a large portfolio of products: if Cloud and DevOps native environments can realize the opportunity to simplify component upgrades we should be able to minimize today's end of life issues and limit the number of central engineering skills they need to maintain internally. But short term, cloud represents another set of products which need to be managed and integrated on top of the existing legacy debt
2. There is a lack of automation and standards: CICD and Infrastructure-as-Code automation are helping us move away from treating infrastructure as pets instead of cattle and will vastly reduce the need for application-aligned engineering and production support, instead folding these responsibilities into Agile teams following DevOps methods

As organizations get further into cloud adoption, they need to be conscious of where ownership will transition for some core control functions provided by Infrastructure teams today. While many of these controls have translated into high-friction processes over time, central functions will still be needed to design and monitor the automated guardrails that are being built around DevOps pipelines e.g.:

- **Security & compliance:** infrastructure teams today provide much of the discipline needed to fulfill regulatory commitments for technology management including but not limited to change management and hygiene. While regulators are not yet providing prescriptive guidance for fully automated environments, it would be a mistake to think we can move all these controls into business-aligned IT

- **Operational Resilience (OR):** in the mainframe and distributed technology eras, infrastructure teams have controlled resilience through data center pairing, replication, load balancing, clustering and failover. As regulators increase their focus on OR, financial institutions need to maintain or even increase their levels of discipline despite control largely shifting to the application layer.
- **Product strategy and standards:** infrastructure has always had a difficult but key role driving the standardization of platforms and products to foster architectural consistency, skills alignment, economies of scale, supportability and end of life risk. In a fully Agile and Cloud-native model it becomes much easier for teams to run their own PoCs and switch product dependencies but there is still value in assessing these shifts centrally for many of the same reasons as before. Enterprise architects play a vital role to ensure that strides toward agility are unaccompanied by a runaway sprawl of the technology portfolio.
- **Enterprise Architecture (EA):** architecture has often been overlooked in agile transformation, being misinterpreted as a figment of waterfall delivery. Mature EA plays an essential role in ensuring that business service architecture can be streamlined with IT service delivery. Without this front-to-back continuous evaluation of business services and evolving technology capabilities organizations will find themselves with a runaway sprawl of technologies, duplicative competing platforms, and mismatched alignment of IT spending.
- **Financial control:** infrastructure has been a conscious bottleneck to control costs (e.g., through available capacity and long lead-times), an important part of annual budgeting for the enterprise and a useful central lever to lower costs during down-cycles. Cloud provides a great opportunity to align costs (via shifts in consumption) with business cycles, but it also needs managing to prevent run-away spend.



Infrastructure groups should re-align toward a customer-centric model to retain their relevance and avoid disintermediation as app teams become cloud-native.

What is the Plan-Build-Run ('PBR') model and why is it no longer optimal?

PBR was designed for large enterprises to promote organizational efficiencies. It aimed to avoid the duplication of functions, to break down traditional IT silos and to foster consistency across regions. The model worked well for firms with slow moving annual planning cycles but struggles with dynamic digital transformation and has accountability gaps leading to user experience issues and friction with Agile development teams.

PLAN

The Plan functions are performed by Relationship Management and Product Management teams who respectively build a picture of business demand and in concert with Enterprise Architects define product and service roadmaps to meet that demand. Together, they delivered a 3- or 4-in-a-box capability (depending on how the role responsibilities were aligned) that comprised business analysis, data analytics, architectural strategy and product strategy. They also worked hand in hand with central project portfolio teams to understand the impact of and organization's large scale change initiatives. The model worked well where strong players ran these functions and built tight business relationships. More frequently though, these functions became a low-value abstraction between engineering teams and their end users; and the product management role is a frustrating one with little ability to control quality, timelines or experience.

Proper planning shouldn't be devalued in an agile model—this is where Enterprise Architects continue to play an integral role—but the structure, frequency and client interaction do change as described in the Customer-Centric section below.

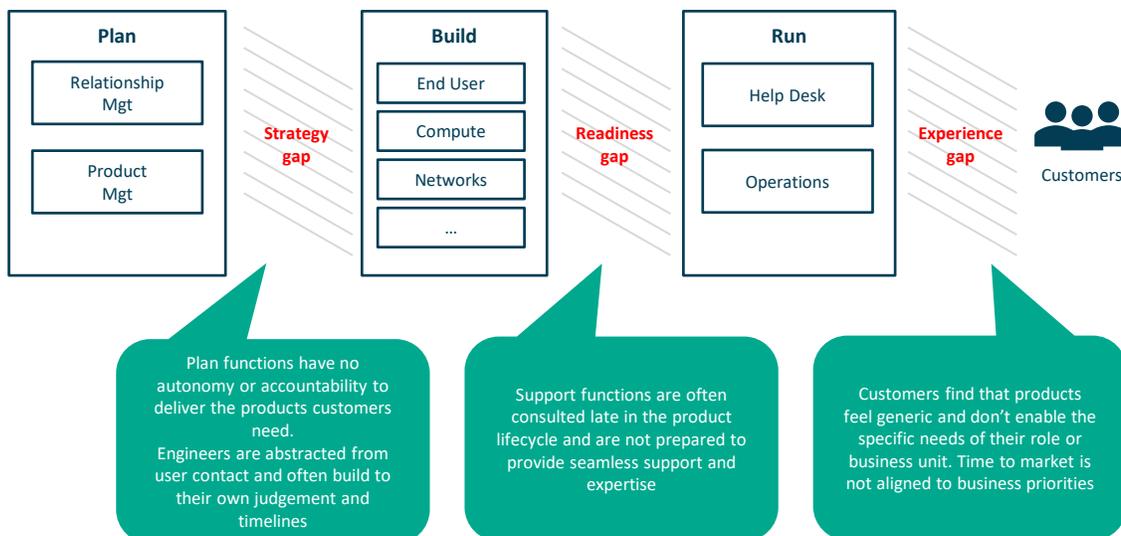


FIGURE 1: GAPS IN THE PLAN-BUILD-RUN MODEL

BUILD

The primary Build function is a cross-technology engineering function. In theory, the model was designed to minimize the inefficiencies of technology silos by having a central engineering group but in practice, these engineering functions are split into silo'd teams anyway (networks, compute etc).



Digital transformation and Agile are re-emphasizing the need for strong engineering cultures with tight business feedback loops. Having strategy and requirements gathering performed outside of the engineering group has always been a source of tension and the disjointed nature of the PBR model is in direct conflict with the agility needed in the digital era. When steered with a continuous view of EA, engineering benefits from a clear view of business and technology objectives as the goalposts for success.

RUN

The Run functions include the operations teams. One big plus of the PBR model was to provide global, cross-technology support and crisis management functions. The common complaint, though, is of operations teams not being involved early enough in the product lifecycle and not being operationally ready or having the necessary expertise when products formally or informally hit production – issues that can be partly solved by good product lifecycle governance, but they are exacerbated by any weakness in relationships between Engineering and Operations functions.



The whole DevOps movement was introduced to improve collaboration between operations and development (or engineering) teams, recognizing cultural gaps that needed to be bridged. While not all infrastructure products are ready for DevOps, the cultural problem is still evident.

What is the Customer-Centric Model?



A customer-centric organizational model is structured around major customer experience groupings – we could just as easily refer to it as ‘Experience-Centric’. At the macro level for most infrastructure departments, this aggregates to two branches:

1. Employee experience: describing the experience of employees and other workforce members using core enterprise technology tools including devices, mobility, productivity and collaboration.
2. Developer (or DevOps) experience: describing the experience of building, deploying and operating applications.

To achieve full accountability for customer experience, product groups must take end-to-end ownership for customer relationship, strategy, engineering and platform operations. This ultimately gives product groups the full autonomy to transform the way its customers work.

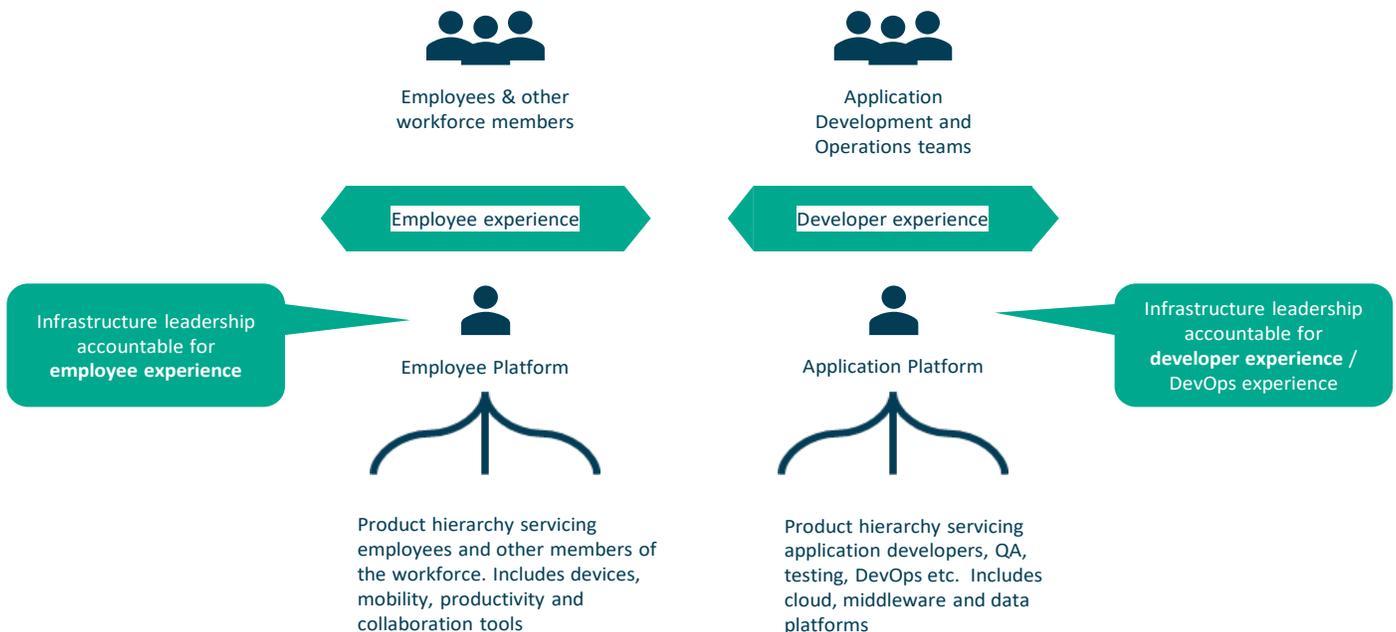


FIGURE 2: PRODUCT HIERARCHY ALIGNMENT TO PROMOTE BETTER END-TO-END EXPERIENCES

If this just sounds like a typical Agile tribe or fleet model, you’re right, in essence it is and that’s exactly the direction infrastructure teams should be headed. Eventually infrastructure will largely be managed as a portfolio of software products and services in exactly the same way as business applications.

We are promoting organizational alignment as well as Agile tribe or fleet structures, proposing that firms have single executive accountability in the two main platform roles i.e. a single head of the entire Employee Platform and a single head of Application Platform while allowing for empowerment and dynamic squads further down the hierarchy. Agile purists may argue for a completely dynamic fleet design overlayed onto functional org structures but we believe there are many benefits in aligning them at a higher level:

1. It provides end-to-end accountability for platforms and experiences, removing the gaps highlighted above in the PBR model.
2. It gives product owners the autonomy to deliver against our goals of building better client experiences and pinpoint IT capabilities and implementation patterns that will drive better client success.
3. It simplifies the ability to put organizational incentives in place, ensuring individuals know what is expected of them and the service value-chain. This cuts down on the misaligned goals and fractured strategies across silo'ed teams that compete for resources today. But organizations must be careful not to fall into the trap of a perfunctory org reshuffle with the same old underlying PBR model and PMO armies. Infrastructure groups need organizational solidity to cope with the co-existence of Agile and Waterfall-based teams. Much of infrastructure's core engineering work involves extensive integration of 3rd party products and semi-manual testing cycles. While this is expected to continue for some time, EA can provide interim strategies that incrementally steer heritage systems towards evergreen replacements.

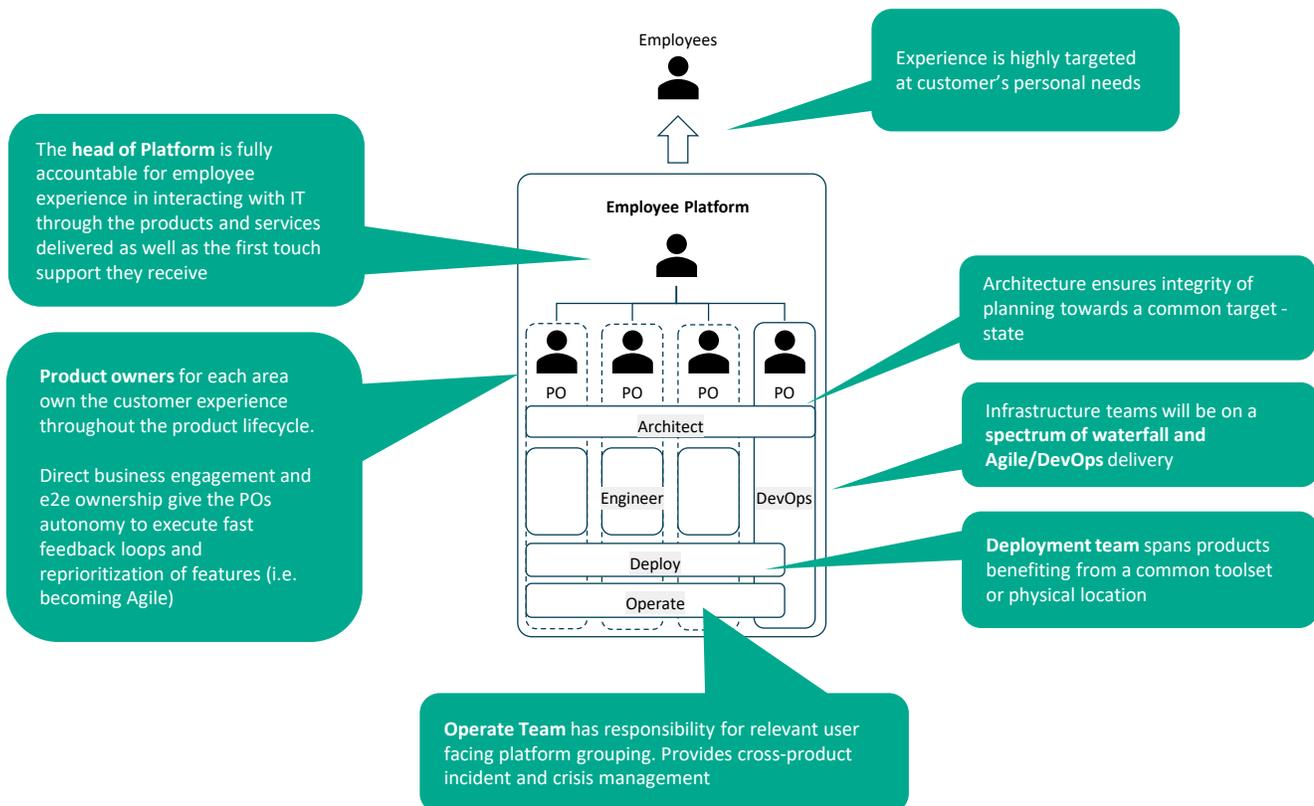


FIGURE 3: THE CUSTOMER-CENTRIC MODEL: EMPLOYEE PLATFORM ILLUSTRATION

What about core services like networks and data centers – how do they fit this model? Over time, internally-managed data center footprints are minimizing and networks are becoming software defined product sets. You can argue that many network services can be aligned to either the employee platform (e.g. WiFi and Voice) or the developer platform (e.g. DC Networks). It is likely that central network teams will exist for some time and are necessary for the dial tone services abstracted away from most developers and employees, but they are also going through a major cultural and organizational shift to software practices.

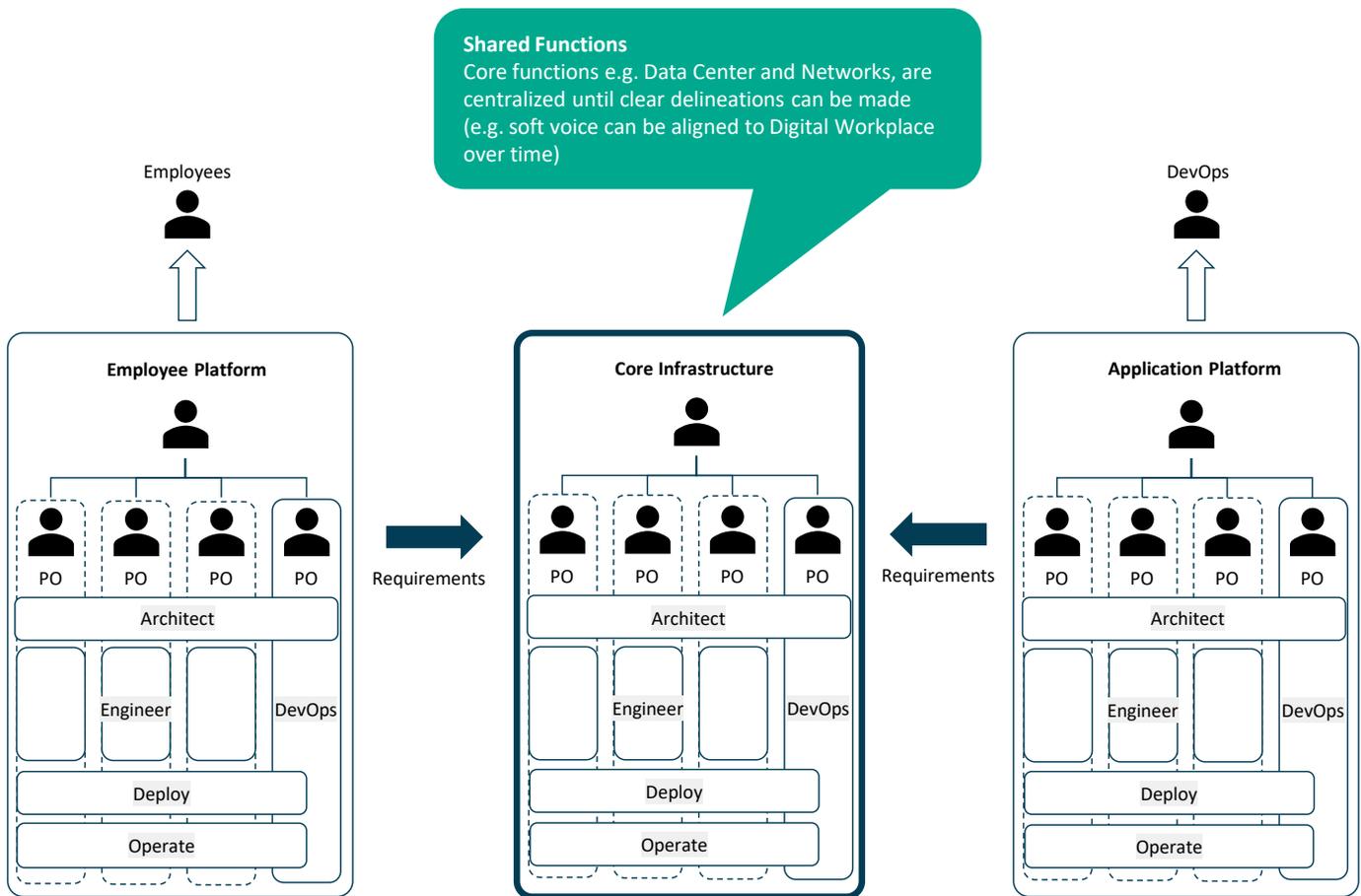


FIGURE 4: CORE FUNCTIONS

Are you on the right track?



If your leadership is still talking about “plan, build, run” you know you’ve got issues. Other signs that you’re on the right or wrong track are listed below:

Signs that you’re on the right track	Signs that you need to rethink your model
Focus is on customer experience	Roadmaps focus on product upgrades (v1 to v2)
Customers feel like IT services and information are personalized to their needs	Customers want to bypass IT
Squads deliver product	Squads deliver intangible outputs
Priorities are aligned directly to business impact and experience	Engineers dictate their own priorities
Delivery in hours	Delivery in months
You can get what you need without picking up a phone	You’re on hold again
Self-service portals with straight-through workflow automation	Self-service portals are just window dressing over high-friction, disjointed workflows

About Citihub Digital

Recoding the Digital DNA of Financial Services



Citihub Consulting is a global, independent IT advisory firm with deep domain expertise across every layer of the technology stack - from business applications and platforms down to core infrastructure. Our consultants have decades of experience helping clients promote best practice in every IT discipline.

Our Heritage: Depth in Financial Services IT

Citihub Consulting understands financial services. Our consulting teams are expert at bridging business and technology to drive digital transformation, comply with complex regulation, and secure critical systems and information.

Bridging Silos: Driving the Modernisation of IT

IT modernisation demands the ability to bridge diverse functions and technical disciplines as DevOps, Cloud, modern application architectures and cyber security blur historical organisational boundaries. Citihub Consulting's consultants span the full technology stack and can act as catalysts to maximise the value of clients' own specialists.

Success: Enduring Client Relationships

Client success is our success. That's why our clients stay with us (we've had year-on-year relationships with our top 15 clients for an average of 8 years). We're focused on building lasting relationships and clients rely on us to honour our commitments whilst giving them confidence that their most challenging goals can be achieved.

