

**Deloitte.**



# Modernization and Product Alignment

Accelerating improvement and adaptability in  
banking and brokerage

# Introduction

If the amount of time and money spent on banking and brokerage technology over the past 20 years was any indication, their applications and websites should be the most user-friendly ever developed. Of course, the opposite is closer to the truth. Why is so much effort resulting in comparatively little progress? And what are some firms doing to reverse the trend?

Technology-driven change has transformed consumer services, with banking and brokerage as no exception. However, the digital services provided by these firms (including, although perhaps to a lesser degree, those from the largest banks) have consistently lagged behind other consumer service-oriented industries. There are many possible reasons for this, but a major culprit is that many firms still rely upon on-premise mainframe infrastructure. Although several large firms are now making progress toward public cloud, core processing remains mostly on-premise across the industry. This legacy infrastructure, in turn, can make it virtually impossible to truly adopt the principles of agile software development, since large swaths of data and back-end processes are by necessity out of the control of even the best product owner. While the financial services industry remains protected from wholesale digital disruption by a large moat of regulatory requirements, the result is that operations, despite decades of effort and the availability of both off-the-shelf software and cost-efficient outsourcing services, typically feature a greater or lesser number of manual, swivel-chair, and even paper-based tasks. Likewise, customer-facing applications, both mobile and browser-based, lag behind customer expectations and often require phone interaction for certain scenarios or to fix issues, despite this being a major red flag for the new generation of investors.

Other consumer services, such as social media, music, video streaming and online shopping, feature applications that are so user-friendly they can become addictive. Services like navigation, rideshares, takeout, and even peer-to-peer payments are, if not addictive, at least majorly supplanting the manual alternative. (Who uses a paper map or writes a check to a friend these days?) Much has been written about the new ways of working that can lead to digital success: agile delivery, the product mindset, dev/ops, continuous integration and continuous deployment (CI/CD), etc. and the exponentially shortened delivery cycle they can bring about. For the layman, this is the difference between your banking app, which releases an update once a quarter to fix bugs and roll out new features, and your video streaming app, which revises its code behind the scenes more or less continuously to correct problems before you notice them and release new features with controlled testing to statistically determine what users prefer without even having to ask.

The benefits of agile development and the path to adoption have been discussed in this space for several years—see Deloitte's previous report on agile banking.<sup>1</sup> Many banks, brokerages, and even asset and wealth managers have ushered in successive waves of organizational restructuring to "be more agile," and CIOs have attempted to prioritize cloud migration and user interface (UI) improvements over this period, leading to dominance among strategic priorities. However, many firms are still wrestling with technical, organizational, and strategic challenges that have prevented gains in technology benefits. Why have so few FSIs achieved comparable results to other consumer services? And what can Financial Services institutions that are still navigating through this effort do to get it right?

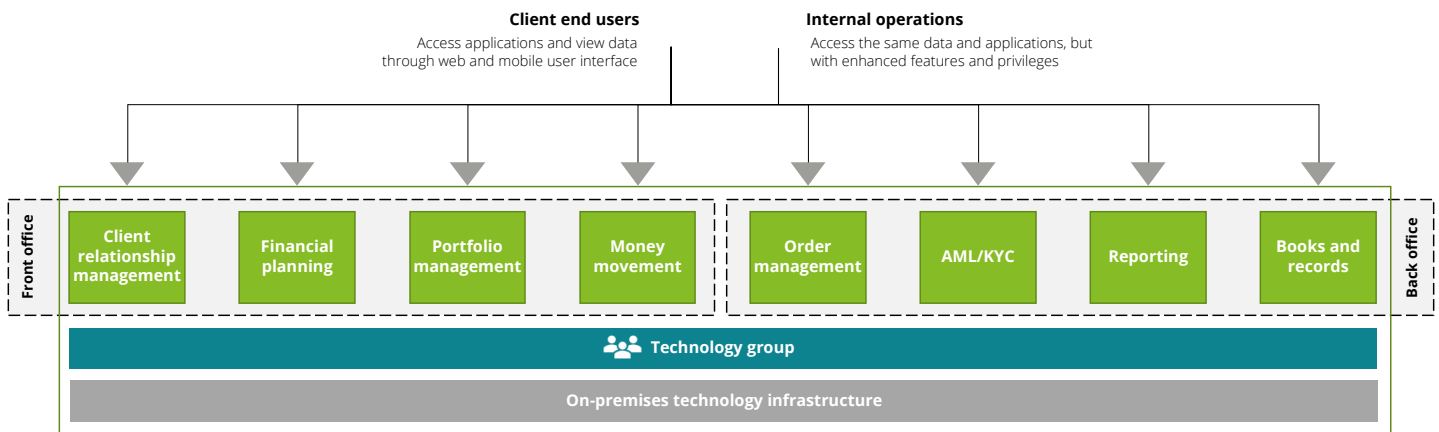
# Modernization

Let us start with the core data that supports the business. Think of accounts, positions, or client demographics; this is generally stored in a monolithic back-end system. Data is freely accessible within the system, for instance, to any application on the mainframe. However, a change to any one component necessitates testing of all aspects of the system, so changes are typically limited to once per quarter to allow time for integrated testing involving multiple teams. Given the advanced age of such systems, accessing their data from more modern systems within the enterprise can also be challenging.

Three broad horizontal categories characterize most organizations: The business forms the first layer, usually defined as the front-office and client relationship management. It is supported by operations, consisting of professionals who operate the company's systems and deal with any exceptions or fallout when automated systems fail. And finally, technology is a kingdom unto itself, consisting of both the technology infrastructure, along with its own operations force to keep it running, the business-facing applications that run on it, and the developers who create and maintain them.

**Before product alignment**, external clients and users from internal operations access front-end applications developed by a horizontal internal technology group, utilizing on-premises infrastructure. Requested changes to front-facing applications must be considered and prioritized collectively, and hardware provisioning adds to the timeline.

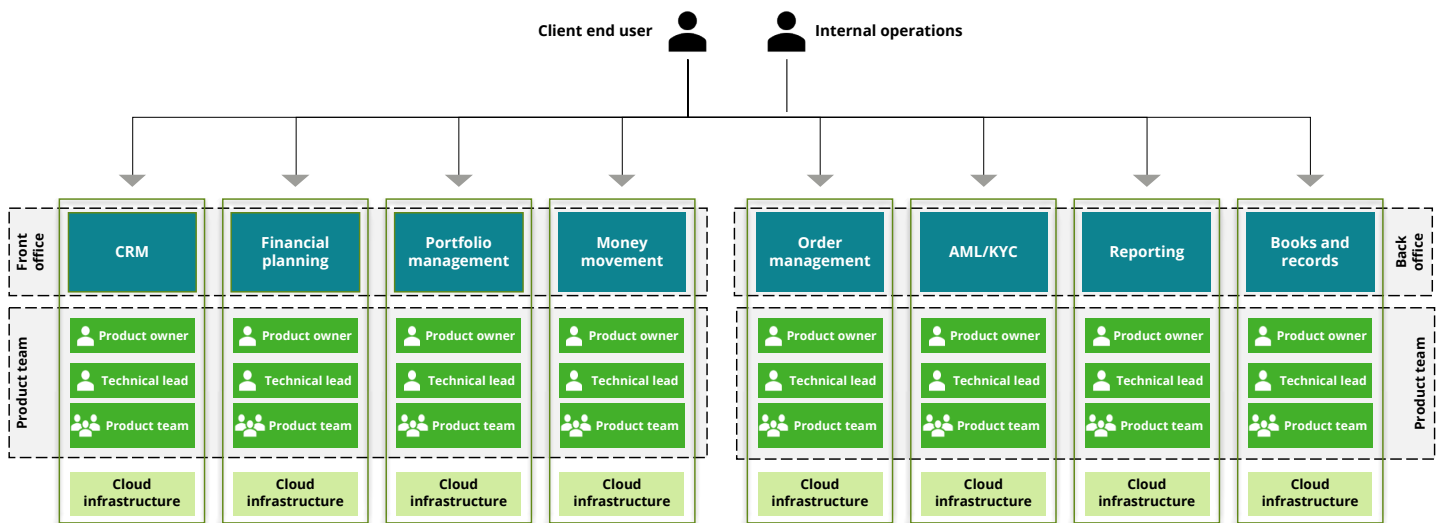
**Figure 1. Before product alignment**



The goal of product-oriented modernization is to break these layers into vertical “products” to enable agile delivery and empower a technical team, aided by a scrum master but ultimately directed by a business representative, such as the product owner, to control a distinct piece of functionality all the way from the user experience, through the operational processes that support it, to the application language and the number of virtual cloud servers it runs on. Products that are bounded into discrete vertical stacks and controlled in this way can make rapid modernization progress, while functions that depend on horizontal swaths for their support simply cannot.

**After product alignment**, each application sits on a discrete “micro” technology stack with its own leadership and supporting team. The product owner and technical lead are empowered to make decisions to change the product to make improvements, solve pain points, and respond to client feedback.

**Figure 2. After product alignment**



Microservices, as opposed to a monolith, feature autonomously owned, loosely coupled components with automated testing and deployment capabilities. They are structured in such a way that components can be changed without disrupting their current functionality, so new features can be tested and added without affecting front-end consumers (e.g., web and mobile pages), who can adopt the new feature, or not, when they are ready. These front-end UI applications often find their scope of modernization and improvement blocked by integration with the back-end monolith, where changes are infrequent and expensive. With microservices, the requests of such consumers are constantly prioritized by the product owner to balance the available capacity of their team against the priority of customers’ enhancement requests, so that the technology is truly responsive to the business.

Public cloud is a big part of the agile product playbook because it frees the team from dependence on internal infrastructure and procurement and enables the automated testing that makes CI/CD possible. We will discuss public cloud further in succeeding sections.

# Making it real

Agile, product-oriented transformation requires a fundamental shift to a new paradigm characterized by development teams having autonomy to determine their own workload. This means that the product owner in-charge of the team doing the work determines when they have completed a given task (e.g., product modernization) and when they are ready to start on the next task (e.g., the next product or enhancement) —described as “pull” rather than “push” methodology. Needless to say, this can be a cultural dislocation at many firms; it requires concerted effort, retraining, and perseverance to overcome cultural resistance. Our experience helping banks and brokerages “make the leap” has revealed several lessons learned that can help firms on their modernization journey.

## **Data is where the rubber meets the road.**

- From a technology perspective, financial services operations consist of retrieving and changing data.
- Bringing data out of the mainframe and making it accessible to next-generation applications on cloud enables agile delivery.
- In order to improve an application or operation, establishing data ownership is key. Improvements will only be possible when the product team has ready access to, and sufficient control over, the data needed for the operation.

## **Expect resistance to new ways of working.**

- Agile delivery often requires breaking objectives into minimum viable products (MVPs) and starting work in parallel.
- Rework is common in the early phases of delivery.
- The goal is not to deliver on a set of objectives; the goal is operations and technology that can change quickly to support new business objectives that are not yet known.
- Antagonism between operations/business and technology is now even more counterproductive; they can only be successful if they trust and listen to each other.

## **Prioritization is a two-way street.**

- Leading practice is to prioritize based on a combination of business impact, technical feasibility and level of effort.
- Program leaders and product owners must optimize resources and output to maximize achievable benefits, and setting an effort threshold can help; even the highest impact objective must be assessed for level of effort and feasibility. Maybe the same effort could achieve many smaller objectives with collectively higher value.

**Product owners should be eager to make progress.**

- The best product owners not only have knowledge and experience but also a relentless curiosity about their products, their consumers, and the supporting technology.
- A desire to constantly improve the product, exceed customer expectations, and employ better technology is tempered only by the team's available time.
- Product owners who balk at the next challenge or refuse to take on additional products or enhancements may not have sufficient resources.

**Documentation is a common pain point.**

- Many organizations don't know how good (or bad) their process documentation is until they try to use it.
- Poor documentation means more time is needed from key business subject matters specialists to deliver modernization, which quickly becomes a bottleneck.

**Pay special attention to third-party integrations.**

- Third-party technology and outsourced service providers (vendors) will operate on their own schedule.
- Any expected (or unexpected) dependencies need to be identified as early as possible and carefully monitored.
- Product teams that depend on third-party offerings may be required to work around their limitations.

# Outsourcing

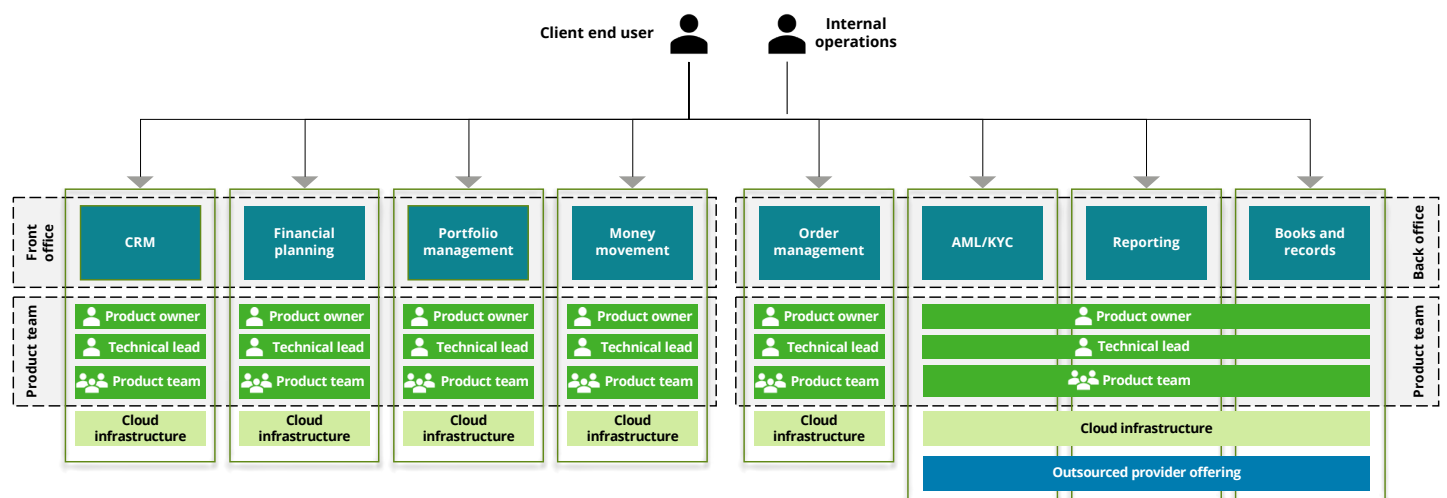
A discussion of operations and technology in financial services would not be complete without acknowledging the prevalence of outsourcing, because what organization doesn't have a third-party provider somewhere in their footprint or, indeed, an outsourcing or insourcing play in progress at any given time? Outsourced providers can reduce fixed costs and allow firms to focus on their bread and butter: building client relationships and offering financial services. However, to a modernized, agile product, they can seem like stones in the path of a swiftly flowing stream—everyone must adapt around them. This is because they themselves generally run cost-efficient yet cumbersome mainframe systems with limited scope for change or customization. An outsourced provider linked to the in-house mainframe for processing is a relatively inflexible arrangement that can be doubly challenging to pick apart when changes are required.

Across financial services, outsourcing providers have not only grown in scale and complexity but also in breadth of services. In many cases, outsourcing providers become central to day-to-day operations; they are the golden source of books and records, produce statements, perform custody and asset servicing, or calculate complex cost basis and performance. They also represent an opportunity to use white label arrangements and expand beyond the firm's core capabilities without taking on the overhead of a new business line; for example, an asset manager that offers private banking or supports small business retirement, or a broker who uses a third party to expand into niche product offerings. Vendors offer reliable services proven across the industry, at favorable comparative production support costs. These providers take responsibility for cost, talent, and risk in what are generally seen as commoditized services; and serve as a trusted partner to handle complex operations.<sup>2</sup>

With outsourcing, impacted products will have less processing work to do, as the vendor partner picks up the slack. Integration into data systems and user interfaces is still needed, but one product team can likely specialize in vendor interaction and support several outsourced products.

While outsourcing can have obvious benefits, firms must be aware of the loss in control that could naturally arise. When customization of vendor functions or data is required, this can lead to tedious and expensive projects. Without an agile product team to own these changes, they will likely fall to the mercy of a vendor's timeline and conditions, and even a responsive and innovative outsourcing provider is not fully incentivized to create new and customized offerings for its client's clients.

Figure 3. With outsourcing



# Case study: Breaking a third-party logjam to enable faster cost basis

In our experience, the opportunity for workflow transformation is occasionally limited due to vendor dependencies. In one particular case, we helped our client, a large wirehouse, undergo an organizational transformation to a product ownership model, including a cost basis product. The client maintained dozens of complex batch jobs and data tables to support the processing of an input file from its books-and-records provider, ultimately generating a file for consumption by another vendor who handled cost basis calculations. The client wanted to utilize real-time cost basis services from its vendor, but was blocked because the vendor could only consume the data after it was transformed during in-house overnight processing.

In response to urgent attempts to convince the two vendors to communicate with each other, both blamed their own antiquated architectures. The books-and-records provider pointed to several customizations it had already put in place for our client, which made the client's processing incompatible with the latest version of its service. The situation became farcical when the vendors merged with each other, yet still required the client to maintain its process to massage the data and pass it from one entity to another.

After years of deadlock and frustration, and after shifting to a product ownership model, the product owner made the difficult decision to refactor the data enrichment process to cloud so that it could be processed on demand. This represented a frustrating additional cost to our client, but was deemed worth it because enabled cost basis to be calculated, even for nonstandard lot selections, a full day earlier than before.



# Breaking up the monolith

Because data and processes within a mainframe can interconnect freely, their interdependencies tend to only grow and become more complex over time. Given that these systems are more than 30 years old, untangling their interlinked applications into discrete products can be quite intimidating. A change to one process can affect many others, necessitating laborious testing and creating a chicken-and-egg problem, with teams asking, “how can we modernize our product when we depend on product X, which is not yet modernized?”

One approach is to replicate the entire system in a modular architecture, test it, and then switch production capacity over once it is ready. The resulting “monolith in the cloud” can then be more easily redesigned into products because all of the data from legacy is now available in cloud architecture. A drawback of this “big bang” method, however, is that it does not start to deliver business value until the massive cloud conversion is complete. Even the legacy system being copied would likely change over such a long time frame, and the business requirements it supports are changing even faster. The program can therefore lose momentum (not to mention funding) and collapse before it starts delivering value.

An alternative is for the system to be broken up into discrete business functions, or products, which can be separated from the whole and delivered as completed functionality in the cloud and then back-synced to the mainframe so that adjacent functions waiting their turn to modernize continue to function as is. This approach means that value can be delivered as rapidly and discretely as possible in true agile fashion, which gives the program early wins and the momentum to carry it through the long road ahead. Early wins can be important because in both this and the big bang approach, costs will increase while the modernized systems are developed and run in parallel with legacy. The drawback to this approach is that the overall timeline could be longer, and legacy systems will have to be supported for longer, than with wholesale replication to a “cloud monolith.”

# Public cloud

With the advent of public cloud, the data center and its supporting staff—the ultimate horizontal organization—is no longer necessary. Now, a small team can make decisions to scale up and down their infrastructure on the cloud, tailored to the needs of their product alone. Depending on the efficiency of the data center and the computing capacity versus business need it provides, this may not represent a massive cost savings—but it does benefit from converting from a large, fixed upfront investment to variable cost and paying only for what is needed.

Even in the relatively conservative financial services industry, public cloud infrastructure is rapidly gaining traction as the method of choice. In 2023, 48% of banking and investment services CIOs surveyed by Gartner reported having deployed workloads to cloud, and 44% planned to deploy cloud workloads in the next 18 months.<sup>3</sup> However, experience across financial services reveals that although organizations may have cloud deployments, the bulk of their processing remains on monolithic mainframe applications that date from the 1990s or earlier.

The cost of the foundational modernization required to address these systems is high, while the cost to maintain a fully depreciated mainframe can seem relatively low. Client-facing sites and apps provide baseline functionality so pressure to modernize can be indirect, and therefore many institutions are not truly invested in moving their fundamental data processing to a cloud-first model. Nevertheless, the cloud migration trend is gaining steam: In 2021, 29% of banking CIOs surveyed stated that they intended to reduce data center technology funding over the coming year. By 2023, that number has risen to 47%, while in the same year 53% of CIOs expected to increase cloud technology funding.<sup>4</sup>

The speed of cloud deployment is what makes the difference for business agility. Instant availability of additional infrastructure capacity, combined with CI/CD toolchains, enables rapid development and deployment that makes the mainframe process, with its quarterly release cycles and monolithic test processes, seem to stand still. The CI/CD toolchain allows changes to application code to be thoroughly tested through automated processes and deployed across environments in only a few weeks.<sup>5</sup> Suddenly, bug fixes, improvement requests, and integrations can be worked through iteratively, resulting in rapid release of new and better features and a dramatic overall improvement in the client-facing experience.

# Product ownership

Prior to starting the journey to transition from a monolith to a product organization, a firm must determine which of its offerings merit designations as “products.” The following guiding questions help make that determination:

| Guiding question  | Example   | Counterexample  |
|---|---|---|
| Does the offering serve a key client function?  | Client bank account details (at a non-bank brokerage firm):   | Client data: Data, including demographics, contact info, and client preferences, is too large and |
| Is the offering compact enough to be owned by one team, yet large enough to warrant a dedicated team? | A centrally stored, encrypted archive with validation functions and an API to read or edit/delete the details based on client ID or account number. | has too much diverse functionality to represent a single product and should be broken up further. |
| Can the offering be considered an atomic unit of the business (i.e., can't be broken up further)?     |   |   |

Note that the role of product owner is a challenging and critical cross-disciplinary role, where successful practitioners have, or develop, expert knowledge of the business function and its underlying technology. A few individuals (product manager, architect, design lead) should be able to work across the teams to ensure collaboration without inhibiting the team’s autonomy.<sup>6</sup>

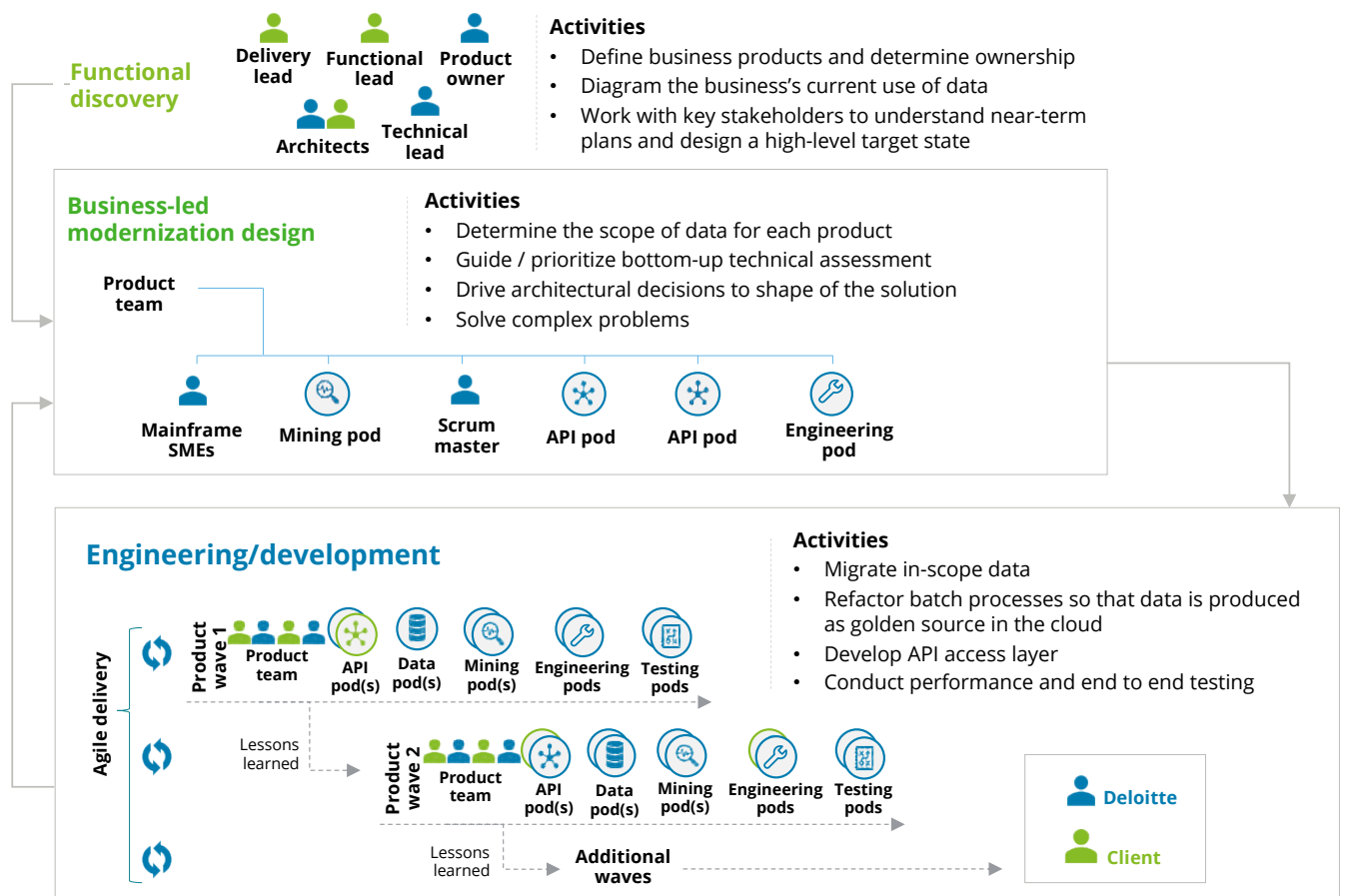
A major goal of the modern organization is that processes should be straight-through and seamless, including interaction with the provider. That means the team managing the product is small for the volume of activity (e.g., trades, balance inquiries, client data maintenance requests) flowing through it from websites, mobile apps, and operations workstations. Agile team members enhancing and maintaining each product, therefore, have to become the go-to subject matter specialists for the firm.<sup>7</sup>

# Case study: Brokerage modernization encompassing 20M+ lines of code

Our client, a large and growing retail brokerage, embarked on a large-scale modernization for its brokerage platform. Prior to starting the journey, we worked closely with the client to divide the platform into 18 high-level “product families” corresponding to key functionality across the brokerage and customer data domains. This would allow our client to not only accelerate its adoption of the product mindset, but also break up the modernization into smaller chunks to deliver a stream of early wins.

In order for the client to achieve its goal of business-led modernization, combined Deloitte and client teams first engaged in “functional lab” planning, including product vision, high-level business architecture, data identification, and finally technical architecture and planning. Each product team then continued with its products until achieving cloud as the golden source for its data, with several “waves” of products modernizing in parallel. This changed the client’s massive mainframe footprint, featuring ~20 million lines of code, ~9,000 tables, ~10,000 batch jobs, and ~2,000 stored procedures into a set of understandable and achievable business-oriented products.

Figure 4. Operating model



One of the products that was prioritized was a portfolio analysis tool. The primary challenge facing this product was that its data resided in tables that were shared with other products—so there was no single technical lead with sufficient expertise and knowledge on how to decouple the data for the migration. Further, the product owner was part of the team that owned the UI and was very familiar with the function, but not the back-end data.

Working closely with mainframe architects, a design to identify the necessary data in the shared tables came first, which could be loaded to cloud to serve as a baseline. Next, the team developed a set of microservices to replace the old java and stored procedure calls and allow customers to set up and modify their portfolios through the website and mobile app.

Due to the product-oriented operating model, architectural decisions were driven by product requirements, with the product owner taking technical input but ultimately owning all decisions. This meant that the product owner had to get up to speed quickly to understand the pros and cons of all technical decisions and build the confidence to know when the company was making the right decision for its customers.

Previously, portfolio analysis was only able to integrate with accounts and positions data during the overnight batch cycle, leading to a clunky experience where clients could only select accounts and positions for analysis if they had been present the previous day. This led to constant support calls by customers who concluded that the tool must be broken, but they had to be told to wait until the start of trading the next day before setting up their portfolio.

By disentangling from the original mainframe monolith, the team was able to achieve real-time integration between the portfolio analysis, accounts, and positions. This led to very tangible customer experience improvements because now trades were reflected instantly in the portfolio, which resulted in a drastic reduction in support calls. Mainframe disentanglement was the biggest hurdle and was achieved over the course of eight months: three for planning and five for engineering and migration. Now, however, the team is able to release code for incremental product enhancements every three weeks.

# Conclusion

Having a product-oriented organization is critical to delivering responsive customer-facing applications. It depends on an agile software development organization, which, in turn, depends on cloud adoption, breaking up monolithic systems, and making fundamental back-end data readily available. Brokerage and banking organizations that are able to embrace the challenge and reorganize around their core products can take advantage of the latest technology, become responsive to their customers' needs, and recruit the best and brightest talent. They are also able to develop the momentum, quality and capacity for improvement and innovation that becomes obvious to their clients and competition. For those that feel they can continue to coast on established reputations, existing infrastructure, and efficient but static offerings, they may be correct—but for how much longer?

# Authors

## **Tushar Daru**

### **Principal**

Core Business Operations  
Financial Services  
Deloitte Consulting LLP  
[tdaru@deloitte.com](mailto:tdaru@deloitte.com)

## **Phil Matricardi**

### **Senior Manager**

Core Business Operations  
Financial Services  
Deloitte Consulting LLP  
[pmatricardi@deloitte.com](mailto:pmatricardi@deloitte.com)

## **Becca Stewart**

### **Consultant**

Core Business Operations  
Financial Services  
Deloitte Consulting LLP  
[becstewart@deloitte.com](mailto:becstewart@deloitte.com)

## **Thalia Rich**

### **Managing Director**

Core Business Operations  
Financial Services  
Deloitte Consulting LLP  
[thrich@deloitte.com](mailto:thrich@deloitte.com)

## **Ricardo Martinez**

### **Manager**

Core Business Operations  
Financial Services  
Deloitte Consulting LLP  
[ricamartinez@deloitte.com](mailto:ricamartinez@deloitte.com)

# Endnotes

1. Daru, T., Sathe, V., & Geetika, A. (2020). [Fast and frequent delivery](#). Retrieved from Deloitte.
2. Cullen, M. (2022, April 11). [Wealth Management Outsourcing: Process Process as a Service \(BPaaS\) - Insights | FIS](#). Retrieved from FIS Global.
3. D'Orazio, V., Malo, J., & Other. (2023). [Cloud Computing User Cases for Banking and Investment Services](#). Gartner.
4. D'Orazio, V., & Malo, J. (2023). [Core Banking Hot Spot 2023: Moving the Core into the Cloud](#).
5. Sacolick, I. (2022, April 15). [What is CI/CD? Continuous Integration and Continuous Delivery Explained](#). Retrieved from InfoWorld.
6. Swanton, B., & West, M. (2023, March 7). [Break Through the Barriers to Scaling Agile Product Delivery](#). Retrieved from Gartner.
7. Wong, J., Murphy, T., & Mann, K. (2020, October 21). [Strategic Roadmap for Becoming a Digital Product Organization](#). Retrieved from Gartner.

# Deloitte.

## About Deloitte

Deloitte refers to one or more of Deloitte Touche Tohmatsu Limited, a UK private company limited by guarantee ("DTTL"), its network of member firms, and their related entities. DTTL and each of its member firms are legally separate and independent entities. DTTL (also referred to as "Deloitte Global") does not provide services to clients. In the United States, Deloitte refers to one or more of the US member firms of DTTL, their related entities that operate using the "Deloitte" name in the United States and their respective affiliates. Certain services may not be available to attest clients under the rules and regulations of public accounting. Please see [www.deloitte.com/about](http://www.deloitte.com/about) to learn more about our global network of member firms.

Copyright © 2024 Deloitte Development LLC. All rights reserved.