WHITE PAPER

Scaling for Success:

A Roadmap to Application Modernization for Financial Services

Table of contents

Introduction

The rise of fast-moving financial technology firms—or fintechs—has changed how traditional financial services firms behave and the way they think of themselves. Swathes of younger consumers have been shunning the traditional sandstone bunkers their predecessors trusted for neobanks that seem to exist only on their smartphones.

These digital native fintechs realized that the oodles of cash they saved from not having a physical presence on the high street of every city could be redirected into the customer experience—from instant and frictionless cross-border payments to digital chatbots that could do the job of thousands of bank tellers simultaneously.

But as we all know, sophisticated digital tools and products rely on data inputs—and incumbent financial services organizations realized they were sitting on treasure troves of data that fintechs could only dream of.

So now the race is on for incumbent financial services organizations to bring the customer experience into the 21st century before fintechs can acquire millions of terabytes of data.

How do incumbent financial services organizations get there and where do they even start?

First, they have to get the basics right. Every successful transformation is based on people, processes and technology.

This guide aims to give you a practical understanding of how to tackle the at times mystifying approach to modernization, and along the way, you'll find examples of other banks, insurers and asset managers that have already been there.

Get buy in

There are three keys to delivering successful solutions—people, processes and technology. In most instances, the people part is both the most critical element and the hardest to get right.

That's not to say that the technology part is easy, but these days we have sophisticated tools that can address most business challenges fairly effectively. For any solution to be truly successful, it's not enough to be able to address the issue technically—it must do this in practice as well, and that means it needs widespread adoption within an organization.

Usually, this hinges on two key stakeholder groups:

Executive team

Getting the CTO, CIO and even CFO on board is key to any technology project going ahead, let alone succeeding. However, within financial services, this need is even more pronounced.

This is because financial services organizations have typically operated on a project basis, with tightly controlled budgets and timelines. Over the years this has contributed to a start-stop mentality.

This is at odds with the continuous and iterative approach used in modern software development. You've probably noticed that the apps on your smartphone get updated regularly—this is because there is always something that can be improved, and the best feedback your developers can get comes from real people using the live solution (this is also why many companies release apps that are still in development, or beta).

By the same token, the apps your bank or insurance company build should use the same approach, lest they become dated, clunky or otherwise unusable. Executives must be 100 percent bought into this new approach, otherwise the project will be destined for failure from the beginning.

Users

As mentioned earlier, one of the metrics of success is that the technical solution ends up getting used. If you are creating a solution or application to be used internally, that means the employees you expect to use it should be involved from the beginning. Not only can they bring insights that will help your developers design a more appropriate solution, but by bringing them on the journey, you will also create a more meaningful experience which should lead to higher adoption once it's live.

When building a more customer-oriented solution, it might not be practical to have all of your customers involved from the initial planning stages, but it might be beneficial to use small focus groups, where you can receive direct feedback from the end users. Beyond face-to-face (or Zoom) focus groups, there are other ways to get feedback from customers. More tech-savvy and engaged customers may be interested in signing up for the beta version of the app, where they will get an advanced look at new, unrefined features and have the opportunity to share their feedback either through direct communication or usage data. If launching a beta version is not practical, due to customer data issues or for any other reason, once the app is live you will undoubtedly receive feedback from your customers that your product teams can use to make continuous, incremental improvements.

Citigroup recently embarked on a long-term digital transformation to help release products to services more quickly, and compete with fast-moving fintechs. Since working with Cloud Foundry, there has been a 57 percent increase in the speed of development. But to get there, Citi's CEO of Global Consumer Banking observed that the company had to first grow their perspective, skills and capabilities while becoming more open to continuously learning new things. If the executive teams were not bought in from the beginning, efforts to replace incumbent waterfall practices with agile development would have hit a brick wall.

Make sure all the stakeholders for an application are brought into the modernization effort.

Set expectations

Once all the key stakeholders are on board, it's important to set expectations about the rate of progress. Most financial services organizations run hundreds of applications—so modernizing the entire portfolio will require a lot of time and effort.

When taking a modern approach to software development, product teams must also have the freedom to innovate—which in turn means the freedom to fail. In a continuous, iterative approach it is necessary to occasionally take one step back in order to take two steps in the right direction.

With this in mind, we recommend financial services organizations assess their app portfolio to determine where they can afford to take some levels of risk. By this we don't mean the type of risk that could catch the attention of regulators or newsrooms—but apps that won't cause a critical outage, should an issue arise.

Considering the above factors, it's no surprise most modernization efforts start small—typically making changes to peripheral systems, like a mobile banking app, and away from the core banking infrastructure. In addition to minimizing the risk to the broader business, this approach also creates more learning opportunities for product teams early on, helping refine processes before moving on to larger and more complex challenges.

With this rapid feedback, technology leaders will have the opportunity to refine processes and take a more systematic and efficient approach. Consequently, this may adjust the expectations that had already been set.

Provide as much visibility as possible into the time and effort a modernization project will require. Avoid over-promising and under-delivering.

Restructure when needed

Modernization is not just about using new technologies, but also evolving the way teams work to match the iterative nature of modern software development. With a clear roadmap and realistic expectations in hand, consider whether your existing teams are fit-for-purpose. Teams that were set up to develop monolithic banking apps on traditional infrastructure might not be set up in a way that is conducive to the modern approach.

And while the structure of your product team is important, there is no one-size-fits-all template—as your teams work on more projects, you will gain a clearer understanding of the structure needed for your particular organization.

It is, however, important for technology decision-makers to build teams with a strong product owner who can act as the customer or end-user advocate. They must have a deep understanding of customer needs and the ability to articulate these requirements clearly, and definitively answer engineers' questions. It's also helpful to have champions from various teams that can bring a different thinking to the table and recognize problems or solutions others might not.

Outside of individual roles, financial services organizations might find it beneficial to split their DevOps teams by area of responsibility. Application DevOps teams can focus on modernizing online or mobile banking products, while platform DevOps teams can dedicate their energies towards improving the banking platform. But it is important that organizations do not separate DevOps from testing. At VMware Tanzu Labs, we believe it's important to have testing, quality assurance and production support within the core product engineering team.

Northern Trust, a Chicago-based financial services organization with over USD $1 trillion under management, wanted to better serve their clients by increasing the velocity with which they released new products and services. This meant adopting a product-driven approach across the entire IT department, including product owners who drive road maps and develop user stories. Unafraid to change how their teams were organized—even multiple times where needed—these changes enabled them to push code into production 70 percent faster than before, with every deployment completely automated.

Prepare for your organizational structure to evolve as modernization efforts advance, but don't assume the approach other companies have taken will work for you.

Prioritize your portfolio

Banks, insurers and other financial services organizations are running hundreds or even thousands of applications, many of which are integral to keeping society running on a day-to-day basis. It's simply not feasible to try and boil the ocean.

Decision-makers should recognize that not every application will receive the same treatment, and that each app should be considered in terms of both the technical requirements to modernize it as well as the business impact it might have.

If you consider a portfolio of a hundred apps, they will typically fall into one of three buckets:

- **No Change** – Roughly 15 might stay as they are with no changes. This includes apps that are scheduled for retirement, like **Internal Applications for Customer Management**, or perhaps those that are so locked in to proprietary hardware and software that they can't be migrated—a **Branch Queue System**, for example.
- **Replatform** – The bulk of the applications—maybe 40 to 50—will most likely be replatformed for a containerized environment. Often, the goal with these is to gain automation and time-savings benefits by incorporating continuous integration and automating the path to production. For an insurer, this might include **Customer Facing apps built on Java/Java EE, Transaction Processing Systems, Large Reporting Systems, etc.**
- **Modernize** – Finally, perhaps 15–20 of the applications will be candidates for extensive modernization. This is most often the set of applications with the greatest potential to move the needle when they are modernized, such as **Mobile Banking Apps, Internal and External APIs, Internet Banking Apps, etc.**

Systematically review your app portfolio and try to place each into one of these three buckets. When you've done this, it might be easier to see how each fits into the end goals you have in mind for the overall business—perhaps the priority is to modernize one particular functional area, or maybe it's to choose one item that will benefit each stakeholder to ensure everyone sees a benefit quickly and remains bought into the overall process.

When you have hundreds of applications that need attention, understanding how to prioritize the apps in your portfolio will have a huge impact on where you ultimately end up.

Choose the right starting point

Even after assigning a priority level to each app in your portfolio, you'll likely still have tens of apps that require immediate attention. Start small and try to find the app that will have the most immediate impact. After all, the ultimate goal for app modernization is to build experiences that translate into revenue.

For a bank or insurer, this usually means the customer-facing mobile application, as this is typically quicker and represents a lower-risk play, though this is not always the case. Depending on the overall digital maturity of the company, it might make more sense to modernize internal, core foundational processes first and then tackle the external-facing apps once those are in a better place.

Recently, a multinational investment bank was looking to leverage its scale and move into retail banking. Their business analysis showed them the best path to follow was to go to market with an online-only neobank offering, so they chose a strategic tech partner to launch a digital credit card.

As their developers started building out the credit card application, they ran into some challenges on the infrastructure side. The bank's foundational applications didn't seem capable of delivering the key functionalities they knew were necessary to deliver the experience customers would expect—namely self-service provisioning, zero downtime and built-in scale and security.

The bank's app development team worked with VMware Tanzu on a proof of concept for one of the card's core service applications, built on VMware Tanzu Application Service. The VMware team outlined the key success criteria with the bank's team and was able to systematically deliver each component to have a pilot of the credit card ready within three months.

Pick a few small projects that will start you off on the right foot and help you build momentum as you get further into your modernization journey.

Make technology decisions that work for you

You might have noticed that this guide has first discussed the people and process factors needed to successfully modernize a financial services organization. As mentioned above, it is absolutely critical to have the non-technical underpinnings in place before you start fiddling with any tech tools.

The financial services industry is one of the most heavily regulated sectors around, with different regulations impacting every jurisdiction. So, it's important to realize that each organization will have a unique path to modernization. With that said, incumbent financial services organizations are likely to use a combination or legacy infrastructure and modern applications for the foreseeable future, so it's important to choose development tools that can work horizontally across multiple projects and align with your team's capabilities.

Individual applications will typically require some specific architectural decisions. Choose the architectural elements that make sense for your application and your organization rather than worrying about conforming to the full list of cloud native precepts.

When our customer, Travelers Insurance, decided to develop their rating engine using .NET Core 2.x and Steeltoe, an open source project for cloud native .NET microservice applications, the company built out a CI/CD pipeline for deployment from day one. The chosen application architecture used a choreography pattern and the team implemented a new API to make it easier for future applications to interact with the rating engine.

Don't base your technology choices on a viral Medium blog you read from a flavour-of-the-month start-up. Make sure your technology stack helps your business achieved the target state you want to get to.

Breaking down monolithic applications

Most legacy financial services organizations rely on a monolithic platform, like a core banking system which is slow and inflexible by design. The challenge here is that even when a bank's mobile application is built using modern methodologies, if it relies heavily on data from the core banking system changes, it will inevitably be delayed and even clunky.

This is where breaking monolithic infrastructure into separate microservices can help decouple modern applications from rigid legacy systems, and help the organization become more agile.

But this can be a complex task, so we thought we'd share some tips to keep in mind as you embark on your journey:

Don't overdo it

Remember the principles mentioned above—every solution needs to be fit-for-purpose for your organization. The same is true here—it's easy to try and break everything down into a million different microservices, but you must consider what will help your business achieve its goals. When Travelers Insurance started their modernization journey, they decoupled their rating engine into 13 separate microservices before they realized that they really needed to focus on five key services to deliver the right combination of modularity and performance.

Decide where to divide

How you decide to break your monolith down will naturally depend on your business' unique requirements, but a straightforward approach might be to start by separating apps that don't need to interact with each other. Once you've done this, you may then consider dividing them again based on what data they need to access from the core system. The point here is to reduce the need to transfer large complex data structures across applications to ensure the end-user application doesn't bend until it breaks.

Sometimes monoliths are okay

If you find you're unable to conveniently break down your monolith or you find an application that relies on a backend database running on bare metal, it might make sense to leave it as it is—there's no need to create an operational problem you'll need to deal with later on. Many monoliths can be containerized and run just fine, and you can still take advantage of the CI/CD pipeline to streamline development and deployment.

When DBS Bank, the largest bank in Southeast Asia, recognized that their customers were increasingly choosing digital channels, they decided to re-evaluate their offering and consider how they could deliver a more holistic digital experience.

Realising the need to fundamentally re-architect their infrastructure to improve agility within the organization, DBS Bank chose VMware Tanzu Labs to help them redesign the way they worked and replace many of the monolith apps that were reaching end of life. Since then, the bank has achieved:

- 6x improvement in time to market for major product releases
- 10x improvement in lead time and 6x reduction in effort
- Enabled self-provisioning of infrastructure for developers
- Zero downtime in the last two years

Plan carefully to break monolithic applications into more manageable, bite-sized chunks without worrying if you have ticked every cloud native checkbox.

Pick platforms pragmatically

You might have noticed a running theme in this guide—you should only choose people, processes and technologies that will help you achieve your business goals. Developers are inspired by or may bounce ideas off their peers, however, what's good for the goose might not necessarily help the gander deliver a more intuitive customer experience while minimizing currency transfers across borders.

No single platform can fit all needs, so it's important to choose one that works for you. Regardless of the platform you ultimately choose, it's important to remember that it's only one component of the overall environment—you still need to have an integrated end-to-end workflow; from IDE to code repository.

In 2020, Fiserv needed to deliver financial relief to thousands of small businesses at the peak of the COVID crisis. When the U.S. government passed the CARES Act, Fiserv was thrust into the middle of a complex workflow between the federal government and thousands of banks around the USA.

Fortunately, Fiserv had already embarked on a modernization journey with VMware Tanzu's application platform. This meant that once Keith Fulton, SVP and CIO of account processing, built his cross-functional team of 15 developers, they were able to write more than 100,000 lines of code and ship 436 product releases in only 28 days. Beyond the tech excellence his team displayed, the tangible outcomes were phenomenal—the new workflow helped secure 18,000 loans worth more than USD $1.4 billion and keep thousands of people in a job during the height of the crisis.

Base cloud and platform choices on your organization's specific requirements and capabilities.

Next steps

As you start out on your modernization journey, remember to take a practical approach based on your business' goals. Visualize where your business needs are and work backwards to determine the steps needed to get there.

VMware Tanzu has been a trusted partner for countless financial services providers looking to modernize their digital foundations for a competitive edge—from comprehensive branch transformations to real-time payments platforms and streamlined compliance.

Our VMware Tanzu Labs offering works collaboratively with your existing teams to upskill and align delivery methods with your existing apps to effect sustainable improvements.

Schedule a free virtual consultation with the VMware Tanzu Labs team today. Each session is tailored to your unique requirements or technical challenge, and we'll make sure you leave with actionable items that will make a difference.

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 vmware.com

Item No: Scaling for Success 3/21 (Updated 7/23)