HCLTech | Supercharging Progress™

Legacy modernization

Why, how and how not to do it

For banks and financial institutions, technology modernization is the need of the hour. Institutions cannot afford to continue with their

monolith legacy applications and infrastructures and at the same time expect to provide superior digital experience to their clients. Technology modernization is always seen as risky but adopting the suitable approach can largely mitigate the risks.

In this whitepaper, we examine different approaches to modernization and ways to mitigate the risk associated with each approach.

Table of Contents

Need for modernization

Banks and financial institutions across the globe are wanting to move away from their legacy applications to **offer differentiated products** to their customers at **rapid speeds,** provide **superior customer experiences and launch new products** without the need for modifying their IT applications. Several factors lead them to drive towards legacy modernization:

- High cost of Infrastructure and maintenance
- Lack of stability and scalability
- Complexities associated in making changes to legacy systems
- Lack of interoperability with newer technology systems
- Longer time to launch new products in the market
- Aging SMEs and lack of skilled resources in the market
- Potential security and vulnerability issues
- Inability to provide seamless customer experiences

Over the last few years, the financial services industry has seen the emergence of neo banks and fintechs that offer financial services over a digital channel as an alternative to the traditional brick-and-mortar and branch-driven banks. Additionally, fintechs and large ecommerce giants are making inroads into the financial services space by offering payments and lending services to both consumers and suppliers operating within their marketplaces.

Adding to this are increased customer demands — especially from Millennials and Gen Z — for personalized and superior digital channel experiences. This demand is driven largely by the experiences they've enjoyed at the hands of those ecommerce giants, transport aggregators and other service providers — for instance, large e-tailers track customer behavior during recurring interactions and use that data to personalize offerings to their customers. Recurring interactions create more data which companies use to design ever-more relevant and superior experiences.

Modernization or replacement of core applications is the way banks mitigate challenges and meet their objectives.

Flexible, lean core applications built on componentized architecture can yield many benefits for banks, including:

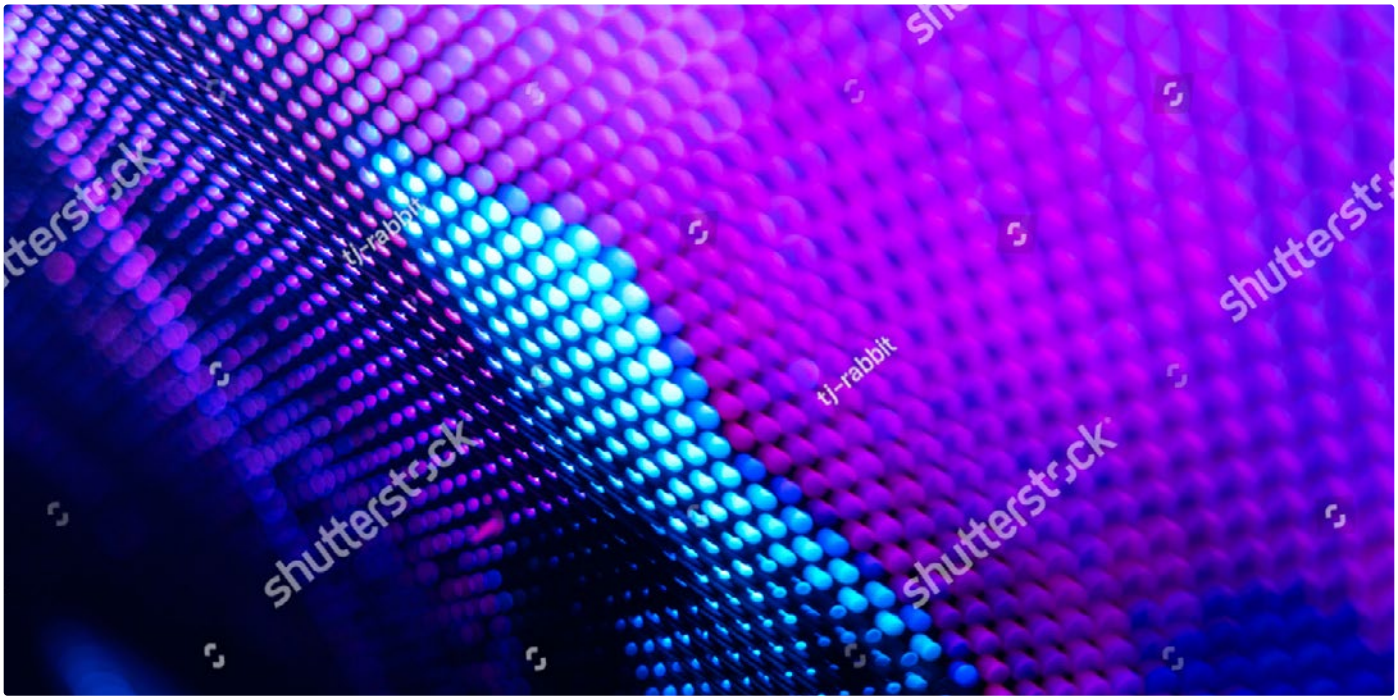
01 Ability to launch new products **faster**

02 Be **better** placed to meet regulatory and compliance requirements

03 Ability to provide **superior** customer experiences and the ability to offer products that differentiate them from their competitors

04 Have **higher** productivity and **lower** cost operations through automation and straight through processing

05 **Easier** maintenance and accelerated adoption of enhancements



Why Modernize?

Many of the legacy systems used by banks and financial institutions are aged anywhere between

10 and 20 years, if not more, and thus they face challenges related to availability of skilled people, process inefficiencies and technology obsolescence.

Despite having compelling reasons to modernize legacy platforms, traditional banks and financial institutions

do not embark upon modernization due to high risk of failure and challenges involved in the entire process of modernization.

Many legacy modernization attempts fail due to lack of support from the business and operations. After all, unless business and operations teams perceive tangible benefits in the modernized application, the program may not have business buy-in and the entire exercise may be treated as just another IT initiative.

Therefore, strong business and technology drivers are essential for legacy modernization programs to succeed.

Drivers for modernization

Business drivers	Technology drivers
Improve time to market and product innovation	Gaining agility in rolling out changes
Simplification of business and operational processes	Overcome skill shortage
Improve customer and colleague experiences	Reduce TCO and ongoing infrastructure through cloud adoption, future ready to adopt cloud, Saas model etc.
Adhere to regulatory and compliance requirements	Reduce code complexity
Reduce cost of maintenance and change	Adopt modern development practices including DevOps continuous delivery
Real-time insights for faster and better decision-making	Move away from disparate, disjoined systems to seamless straight through processes
	Manage and govern data for effective consumption

Before you embark on modernization journey

The road to technology modernization is bumpy, with numerous hurdles that necessitate the need to take appropriate diversions. Therefore, an assessment of the existing landscape is essential to determine the route/ approach for modernization.

Such an assessment helps us formulate a modernization roadmap that considers the needs of the business and IT. The assessment findings and discovery should encompass the state of an existing application(s) from both business and technological aspects.

While the assessment can help in the formulation of strategy, it is essential that the modernization team is agile enough to adapt to course corrections during the journey.

Key outcomes expected out of the assessment:

Business:

- Business products and processes supported by the application
- Future roadmap of business products and processes supported by the application
- Future book of work
- Criticality
- Business value and return on investments (ROI)
- Improvement in customer and colleague experience
- Gaps and inefficiencies
- Compliance risks/issues
- Average term of products in the application (if it is a servicing application)
- Closed book applications being serviced

Technology:

- Complexity
- Technology currency
- Internal and external integration architecture
- Design
- Deployment architecture
- Reliability
- Scalability
- Performance
- Skill availability
- Technical deb

Findings of the assessment will act as input for defining the modernization approach.

Which road to take?

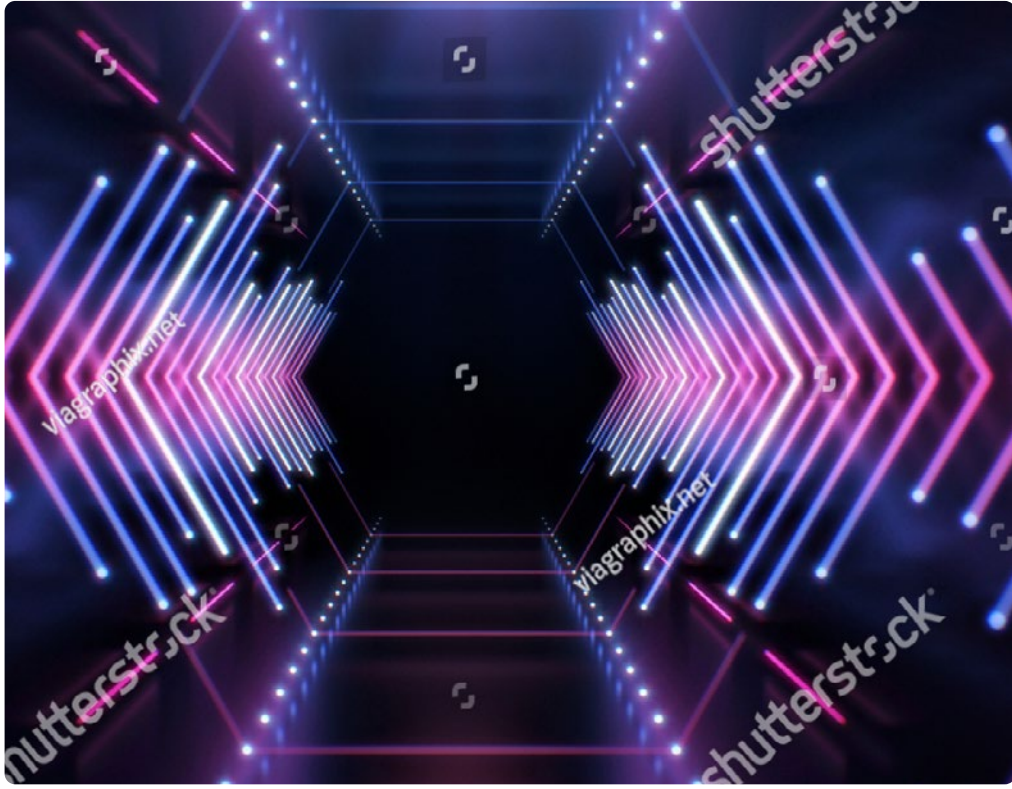
A one-size-fits-all approach to modernization doesn't yield results. In certain cases, there may be a need to adopt a combination of approaches.

Factors that determine the approach for modernization are:

- Key attributes of the application
- Technical and business drivers
- Expected business/technology outcomes

Approach for modernization – the art of the possible

The modernization of an application or set of applications can be planned for tactical or strategic reasons. Broadly, there are 6 possible approaches, with three(refactor, re-host, re-platform) suited to produce tactical outcomes and three (re-architect, replace, rebuild) ideal for strategic and long-term outcomes.



Re-factor

- Technical re-factoring of an application is used when the objective is to:
- Reduce technical debt
- Simplify application maintenance and support
- Modify and fit in a new environment, say the cloud, by making adequate changes

Benefits

Ease of maintenance

No impact to operations

Shorter timeline to rollout

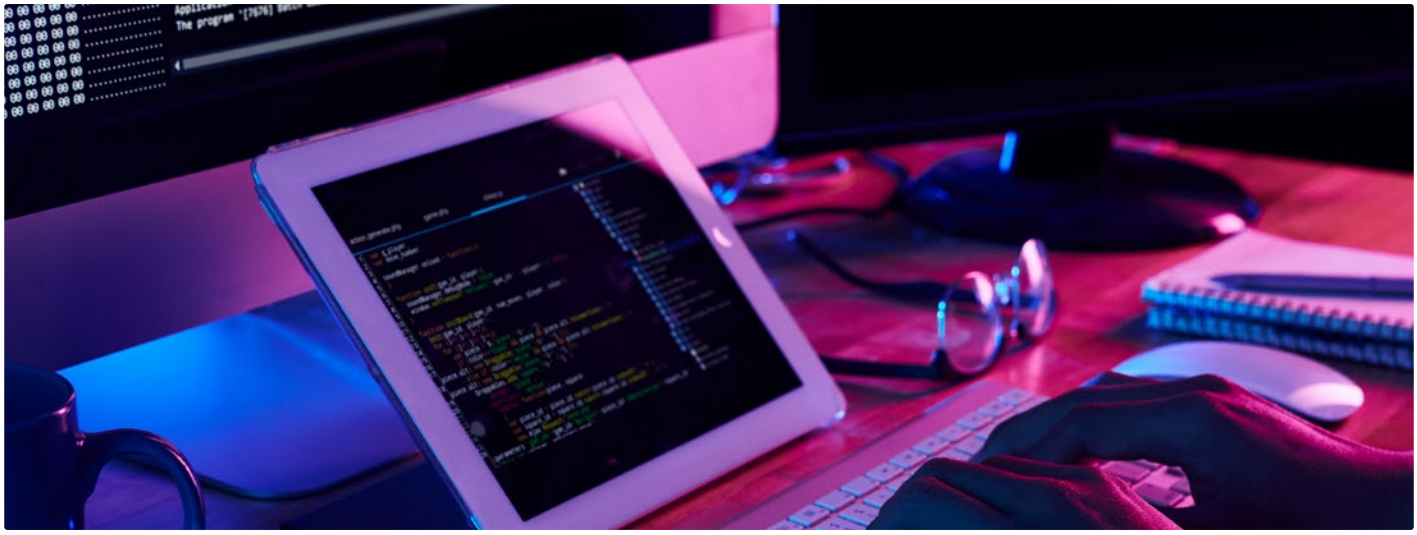
Removal of technical debt

Improved performance

Minimal or no need for data migration

Challenges

- QA approach to validate application features and functionalities
- Completeness of existing documentation, which is essential to validate features post modernization
- Ensuring no loss of functionality



Re-host

Re-hosting of an application is used to overcome infrastructure challenges such as impending end of life for a server.

Cloud infrastructure comes along with not only storage but various other tools (e.g., AI, machine learning, etc.) and frameworks, as well.

Re-hosting of on-premises applications onto cloud infrastructure might help in reducing infrastructure cost.

Typically, re-hosting of the application does not involve major changes to the code or architecture. However, re-hosting may necessitate some changes in the upstream and downstream integrations. The application is moved to a newer infrastructure that requires changes for existing integrations.

Benefits

- Leass risky
- No impact to operations
- Minimal or no need for data migration
- Shorter timeline to rollout

Challenges

- Need for change in upstream or downstream application to enable integration

Re-platform

Many banks do not attempt to modernize their mainframe legacy applications because it is seen as too risky. Micro Focus COBOL has come in very handy in mitigating such risks with the application converted to newer technology without required architectural changes.

A similar approach can be applied to applications built on reasonably newer technologies. Moving to a newer framework helps overcome skill shortages, compliance risks, etc.

It is advised that only essential changes are considered and making changes in-parallel can affect the stability of the application and pose challenges during validation.

Optionally, changes can be made to the application code to match the newer coding standards, handle feature obsolescence, etc.

Benefits

- Low to moderate risk
- No impact to operations
- Shorter timeline to rollout
- Minimal or no need for data migration
- Moving to new version may enable cloud adoption

Challenges

- Rework needed on integration with upstream and downstream applications
- QA approach to validate application features and functionalities
- Ensuring completeness of existing documentation, which is essential to validate features post modernization

Re-architect

The digital era demands that all banking services be available over digital channels for superior customer experiences and seamless delivery of services. Most of the digital services are delivered through APIs and services exposed by core applications. Multiple APIs and microservices are orchestrated to provide services through digital channels.

Legacy applications have been built to handle file-based data transfer and batch interfaces. This limits the bank from offering services via digital channels that require real-time and seamless integration.

To overcome such challenges, the architecture of legacy applications is remodeled to expose legacy functionalities as APIs. Those APIs exposed via an API hub or integration layer are consumed by upstream and downstream applications. The legacy application may continue to be a system of record.

Benefits

- Faster time to market
- Improved customer experience and personalization
- Seamless workflow enable by micro-services
- Leverage data analytics to cross- sell upsell etc.

Challenges

- Need for change in upstream or downstream application to enable integration

Replace

In the post-cloud fintech era, there are many packaged solutions and platforms in the market. Packaged solutions are available as on-premises implementation, PaaS (Platform as a Service) or BPaaS (Business Process as a Service). Legacy applications are typically loaded with features. Therefore, finding a suitable alternative package solution may be very challenging.

One or more packaged solutions may have to replace the entire set of legacy capabilities or subset of capabilities. In order to replace a legacy solution in part or full, banks must follow the below steps:

Define a set of requirements that need to be fulfilled by the target application

Build essential customizations

Evaluate available off-the-shelf product(s) and choose the most suitable

After implementation of the new application, decommission/retire the legacy application in part or full

Replace the legacy application with the chosen application

Benefits

Improved colleague and customer experience

Well-defined support and upgrade plan

Faster time to market

Decommission of legacy system

Challenges

- Resistance to change in ways of working
- Too many customization requirements
- Potential gaps comparing the application against legacy
- Need changes on upstream and downstream applications to support integration

Rebuild

Banks may choose to rebuild instead of other options because of:

- Cost-effectiveness
- Ability to customize as per needs
- More control over schedule
- Incremental and phased movement

Banks may define a completely new set of requirements or retain some and redefine some.

- Adopt state-of-the-art technology
- Re-imagine business processes
- Define future-ready and scalable architectures

Benefits

Seamless workflow enabled by state of art architecture

Improved customer and colleague experience

Can be built as a platform suitable for SaaS model, either on-prem or cloud-hosted

Leverage data, AI/ML, NLP, etc. to enable automation, personalization, differentiated product offerings, etc.

Decommission of legacy system

Challenges

- Technology-driven approach instead of business-driven
- Lack of documentation for existing application features
- Attempting to retain features and ways of working
- Longer time to market and risk of de-prioritization
- Data migration and reconciliation
- Conflicting priorities between application and upstream or downstream application Book of Work (BoW)

Each of the above-given approaches has its own benefits, challenges, factors that influence the outcome of the approach, timeline, cost and associated risks.

It isn't necessary that a particular approach must fit for all types of legacy applications. For instance, consider the following parameters for a legacy application:

- It has minimal or no book of work
- There aren't any compelling needs for business or operations transformation
- There is no risk of technology obsolescence in the near future

Here, the application can be re-hosted on a newer cloud infrastructure to reduce physical infrastructure costs. The application does not require replacement or rebuild options.

Where to host: on-prem or cloud?

One of the key objectives of modernization today is to harvest the benefits of cloud offerings in terms of storage, scalability, elasticity in computing power, as-a-service models and tools, etc. Banks may view modernization as an opportunity to modernize the application and offer it as a PaaS solution to rationalize core applications serving similar purposes across business segments, regions or countries.

The extent of benefits that a modernized application can leverage out of the cloud is based on the adopted modernization approach. It largely depends upon whether:

The application is modernized only to host the application on the cloud to free up on-premises infrastructure

The application is modernized to build it as a cloud-native solution

Modernization approach – on the cloud or for the cloud

On the cloud

Rehost

- Host in a newer infrastructure
- Rework integrations

Refactor

- Reduce technical debt
- Improve performance

Re-platform

- Move to a newer technology framework
- Rework integrations

In most cases, may need implementation in one go

Core Modernization

For the cloud

Replace

- Move to a vendor product/platform
- Rework integrations

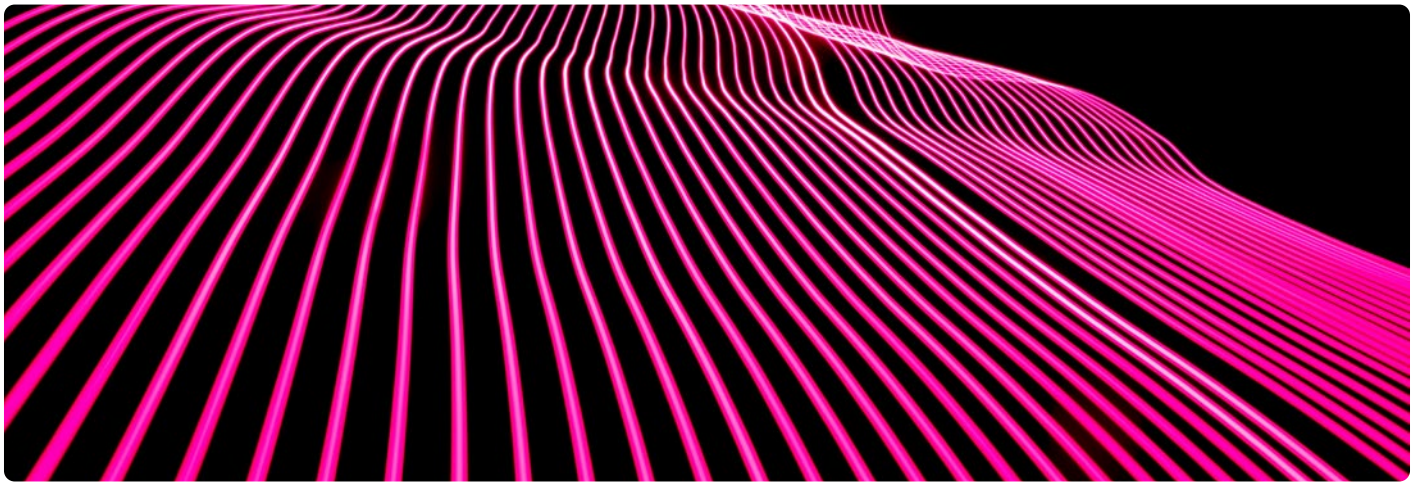
Re-architect

- Move to a newer technology, build API's, microservices
- Retain/rewrite business logic

Rebuild

- Build from scratch in composable, microservices, component- based architecture

Feasible to implement in phases



Which approach to choose?

To determine the approach, evaluate the options from the perspectives of cost, time and associated risks.

If not modernization, banks can have a greenfield setup with a new system built or bought off the shelf (PaaS or on-premises).

These approaches can be broadly classified as below:

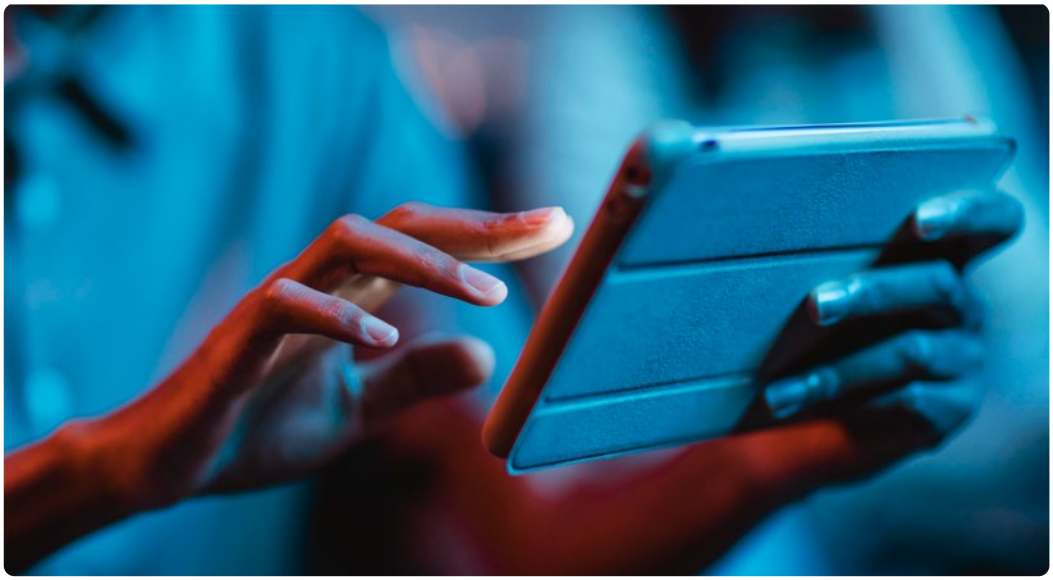
Replace, phased progressive (rebuild, replace, re-platform)

Phased, progressive, iterative replacement (re-host, refactor, re-architect)

Greenfield setup (build or buy or PaaS)

Based on an organization's budget, timeline and risk appetite, the most suitable option will emerge.

Since each legacy application may have its own complexity, timeline and cost, the approach may vary based on each individual application. Thus, we must balance multiple factors before deciding on the final approach.



What are the success factors?

Each modernization approach comes with benefits, success factors and a few challenges. The extent of benefits and challenges may vary based on the approach.

Success Factors

Re-factor	Verify and ensure technology support for the next few years	Legacy application remains fit for the purpose and changes are externalized	Evaluate skill availability in the market	Focus on reducing technical debt
Rehost	Maintain minimal or now BoW for legacy	Legacy application re-mains fit for the purpose and changes are exter-nalized	Ensure continued technology support	Do not embark upon business transformation
Re-platform	Legacy application remains fit for the purpose and changes are externalized	Ensure continued technology support	Make essential changes	Do not embark upon business transformation
Re-architect	Ensure business and operation acceptance	Bring agility to business and IT	Ensure low-medium funding/investment	Cross/upskill resources
Replace	Be open to business and operation transformation	One size may not fit, openness for combination of products/ solutions	Choose best suited commercial model (license or SaaS or any other)	
Rebuild	Ensure business and operation acceptance	Bring agility in business and IT	Ensure medium-high funding/investment	Adopt phased delivery and ensure periodic release of tangible functionality

How do the approaches stand against each other?

Apart from the rebuild and replace approach, none of the other modernization approaches are likely to meet all the business and technological drivers that triggered the modernization plan. However, other approaches may give tactical benefits.

Each individual approach may help banks/financial institutions meetone or more of the objectives/drivers.

	Driver	Re-factor	Rehost	Re-platform	Re-architech	Replace	Rebuild
Business	Improve time to market and product innovation	◆	◆	◆	◆	■	■
	Simplification of business and operational processes	◆	◆	◆	●	⊙	⊙
	Improve customer and colleague experience	◆	◆	◆	●	⊙	⊙
	Adhere to regulatory and compliance requirements	◆	◆	▲	◆	⊙	⊙
	Reduce the cost of maintenance and change	▲	▲	●	◆	■	■
	Real-time insights for faster and better decision-making	◆	◆	◆	◆	⊙	⊙
Technical	Gaining agility in rolling out changes	◆	◆	◆	●	⊙	⊙
	Overcome skill shortages	◆	▲	●	●	⊙	⊙
	Reduce TCO and ongoing infrastructure through cloud adoption, future ready-to-adopt cloud and SaaS models	◆	▲	◆	●	■	⊙
	Reduce code complexity	▲	▲	◆	■	⊙	⊙
	Adopt modern development practices, including DevOps and continuous delivery	◆	◆	■	●	⊙	⊙
	Move away from disparate and disparate systems to a seamless straightthrough process	▲	◆	▲	●	⊙	⊙
	Manage and govern data for effective consumption	◆	◆	◆	▲	■	■

Most/all objectives are likely to be met ⊙

Most objectives are likely to be met ■

Some objectives are likely to be met ●

Minimal objectives are likely to be met ▲

Approach may not be suitable/applicable or may not be beneficial ◆

Author bio

Sivakumar Srinivasan

He is a Practice Director working with the banking and financial services domain practices group at HCLTech. He has 27+ years of experience in business analysis, product implementation, consulting and product management. His domain experience includes consumer lending, mortgages, commercial lending and cash management. He has worked with banks across geographies and has been part of several banking product implementations, legacy modernizations and transformation journeys.

HCLTech

Supercharging Progress™

www.hcltech.com