



Experience the commitment®

Banking

# Application Modernization and Portfolio Management

Key Challenges and Success Factors

As part of their long-term strategic plans, banks are seeking to capitalize on their legacy applications. Acquired through decades of IT investment, as well as acquisitions, these applications represent deep know-how, drive core business processes and support mission-critical operations that differentiate banks in the market.

However, they also present challenges. Legacy systems often operate in silos, hindering “joined-up” IT and the advantages it brings. Many use technologies that have become obsolete, create vendor dependencies, or are insufficient to support the increasing demands of online banking. In addition, they typically lack adequate documentation and are costly to maintain.

These and other “legacy” constraints can make it difficult for banks to keep pace with new technologies, respond quickly to changing market demands and reduce IT costs. However, through application modernization and application portfolio management, legacy constraints can be overcome, enabling banks to fully preserve the value of their IT assets and maximize their legacy application investments. In this paper, we discuss key challenges and success factors for each.



# TABLE OF CONTENTS

MODERNIZATION VERSUS PORTFOLIO MANAGEMENT.....	01
MODERNIZATION CHALLENGES.....	03
MODERNIZATION SUCCESS FACTORS.....	04
Roadmap.....	04
Approaches.....	05
Expertise.....	07
PORTFOLIO MANAGEMENT VERSUS PORTFOLIO RATIONALIZATION.....	08
PORTFOLIO MANAGEMENT CHALLENGES.....	09
PORTFOLIO MANAGEMENT SUCCESS FACTORS.....	10
Methodology.....	10
Framework.....	10
CONCLUSION.....	11



# Modernization versus portfolio management

While legacy applications are highly valuable and critical to bank operations, many have reached a state where one or several factors are limiting their ability to evolve and quickly respond to ever-changing business requirements. Modernization is the process of removing these “legacy” constraints, while at the same time and, just as importantly, safeguarding the intrinsic value of legacy applications.

Today’s modernization paths enable banks to keep their applications intact while simultaneously enhancing them to drive future business growth. Replacing legacy applications is too costly and time consuming. Plus, banks do not want to toss out systems that are working and represent a huge IT investment. Modernization is the answer, enabling rapid, cost-effective, IT evolution with reduced risks.

Modernization is often an outcome of application portfolio management (APM), which involves assessing a company’s application portfolio and making decisions in terms of its ongoing maintenance and evolution. APM is all about continuous improvement and transforming the portfolio to address evolving needs.

With APM, banks acquire an in-depth understanding of their applications, including their functions, interdependencies, business value, required support skills, etc. APM helps banks to align their business and technology objectives and assess the costs and risks of their current application portfolios. It gives them a broad view of their applications, which is essential to making decisions about their future evolution.

Following are key challenges and success factors for both application modernization and APM, as banks look to each to maintain and maximize the value of their legacy applications.



# Modernization challenges

The main challenge with legacy applications is that they are not typically fully understood. Most legacy applications have been developed over many years by different teams of programmers. They are either poorly documented or not documented at all. While you can start by looking at their specifications and user manuals, and even talk to developers and maintenance staff, in the end, the only 100 percent reliable source of information about a legacy application is its source code.

But, typically a legacy bank application is comprised of millions of lines of code that are grouped in modules tightly interconnected in a complex mesh. Simplifying this complexity and breaking it into manageable components is not an easy task.

Decades of software maintenance usually means that the code and data have been copied and shared over and over in different ways. Sometimes, to further complicate things, not even the source code is available, either due to licensing restrictions or because it was lost. As a result, its requirements must be retrieved from usage logs and black box analysis.

Fortunately, specialized tools and services are now starting to appear on the market to provide this type of analysis. When choosing one of these vendors, it is important to assess its independence, as much as its technical ability, to ensure it is not biased toward a particular technology.



# Modernization success factors

## ROADMAP

The first step in modernizing should be to assess where you want to go and where you are now. The second step is to determine which paths are available to get to your destination and which path you should choose. Only after these steps are taken should you begin the journey.

A complete analysis of the application code and data, as well as the business requirements, must be undertaken. This analysis combined with deep knowledge of the latest technologies and tools will form the basis of an evolution plan or roadmap.

The roadmap should define in detail the following:

**Vision:** Outline of business and IT context and future direction, development of engagement plan

**Current state:** Conceptual and logical models of your current business and IT landscapes

**Desired future state:** Conceptual and logical models of your desired future state for business and IT

**Roadmap:** Performance of a gap analysis, development of a benefits dependency network and IT strategy roadmap



**Solution options:** Assessment of leading commercial off-the-shelf solutions, assessment of the four key modernization approaches (i.e., re-host, convert, re-write, replace)

## APPROACHES

Modernization roadmaps often combine a number of options to reach the right end state, as well as to lay the foundation for future evolution. There are four main approaches, each with its own risks and tradeoffs, which include the following:

- Re-hosting and encapsulation of core applications
- Conversion of core applications
- Rewriting core and supporting applications
- Replacing core and supporting applications

**Option 1** is often best when the code is good, the functionality is likely to remain the same, and the bank is seeking quick and substantial cost savings, as well as ease of integration.

Option 1 Pros	Option 1 Cons
<ul style="list-style-type: none"><li>• Immediate cost savings ( up to 80% of operational and maintenance costs)</li><li>• High reuse of existing assets and investments</li><li>• No core changes</li><li>• Low implementation effort, minimal business disruption</li><li>• Depending on complexity, results delivered in 4-18 months</li><li>• Out-of-the-box Web services interface for SOA integration</li><li>• Based on pre-tested and commercial re-hosting platform</li></ul>	<ul style="list-style-type: none"><li>• Retains legacy architecture and programming languages, forcing a continued reliance on legacy skill sets</li><li>• Legacy code remains unchanged (no improvements as default)</li><li>• Standards like BIAN not out-of-the-box</li><li>• May require re-coding where automated tooling not appropriate</li></ul>



**Option 2** is used when the business logic is good, but the legacy knowledge is poor. It is a value-added modernization route that involves bringing in new staff, exposing legacy business processes and enhancing core applications.

Option 2 Pros	Option 2 Cons
<ul style="list-style-type: none"> <li>• No need for continued reliance on legacy skill sets</li> <li>• Typically, 12 months+ of implementation effort</li> <li>• Mainstream technologies and Web services as a default</li> <li>• Maintain same business logic and processes while exposing new interfaces</li> <li>• Depending on complexity, high automated code conversion can be utilized</li> <li>• Between 30-50% of code rewrite cost</li> </ul>	<ul style="list-style-type: none"> <li>• Longer time to achieve cost savings</li> <li>• Without code "re-factoring / analysis," lack of significant improvements to target code and retention of legacy programming flaws</li> <li>• Lack of major changes to architecture and application design</li> <li>• Challenge to identify application/ module boundaries to be converted</li> <li>• Requires good legacy documentation</li> <li>• Without use of automation tools, high cost and high risk to convert</li> <li>• Always requires some re-coding</li> <li>• Standards like BIAN not out-of-the-box</li> <li>• Requires senior legacy and mainstream technology competence</li> </ul>

**Option 3** is recommended when legacy applications are so complex that automated conversion tools cannot be used. It might be the only modernization option for avoiding new customization.

Option 3 Pros	Option 3 Cons
<ul style="list-style-type: none"> <li>• No legacy code and therefore no continued reliance on legacy skill sets</li> <li>• Best of breed SOA architecture and Web services as a default</li> <li>• No legacy programming flaws</li> <li>• High mainstream technologies usability</li> </ul>	<ul style="list-style-type: none"> <li>• Cost savings available only in long term</li> <li>• High implementation effort (up to 36 months plus)</li> <li>• Minimal reuse of existing assets and investments</li> <li>• Typically, high cost and success rates are low</li> <li>• Requires strong, senior mainstream technology and SOA competence</li> <li>• Requires excessive business participation and common agreement</li> </ul>



Few banks consider **Option 4**, preferring instead to minimize costs and risks by using the other modernization routes. However, this option may be a good one for replacing some mission-critical components.

Option 4 Pros	Option 4 Cons
<ul style="list-style-type: none"><li>• No legacy code and therefore no need for continued reliance on legacy skill sets</li><li>• Best of breed SOA architecture and Web services as a default</li><li>• No legacy programming flaws</li><li>• High mainstream technologies usability</li></ul>	<ul style="list-style-type: none"><li>• Cost savings available only in long term</li><li>• High implementation effort and timeframes can be 3-4 years</li><li>• Typically, high cost and low success rate</li><li>• Match to local requirement, low re-use, significant re-writing</li><li>• All effort diverted to core and away from new business services</li><li>• Typically, high vendor lock (maintenance and support)</li><li>• Reliance on scarce vendor-trained staff for duration</li></ul>

## EXPERTISE

A sound and well-designed modernization roadmap will achieve the key objectives of modernization, including increased agility, shorter time-to-market, cost savings and reduced security risks, among others. However, building such a roadmap requires the right expertise.

This expertise begins with a team of highly competent and experienced technicians with a track record of successful projects and impressive returns on investment. The following is also critical:

- Innovative technological approach to modernization supported by a set of proven tools for the complete analysis of mainframe environments
- Established partnerships with all major modernization vendors (while maintaining vendor independence)
- Modernization strategy designed to minimize risks
- Industrialized practices and processes
- Structured knowledge management framework to facilitate knowledge capturing, maintenance and sharing, as well as team collaboration
- Modernization center of excellence, providing capabilities and services that cover all aspects of modernization
- Global delivery and governance

# Portfolio management versus portfolio rationalization

APM is similar to an annual health or auto checkup in which a person's health or vehicle is evaluated, issues are identified, and recommendations for improvement are made. It is vital to effective asset management and continued optimization.

Through APM, a company can effectively gauge the scope and impact of making any changes to its application portfolio. This is because APM provides comprehensive information on interdependencies (i.e., with other applications, with the IT infrastructure, with business functions/objectives). Unless you understand the interdependencies of your applications, a single change can result in unexpected and negative impacts.

APM is broader than traditional application portfolio rationalization (APR). APM is focused on ongoing governance and continuous improvement to drive alignment with business objectives and cost savings. APR, on the other hand, focuses on simplifying the application landscape to drive cost savings.

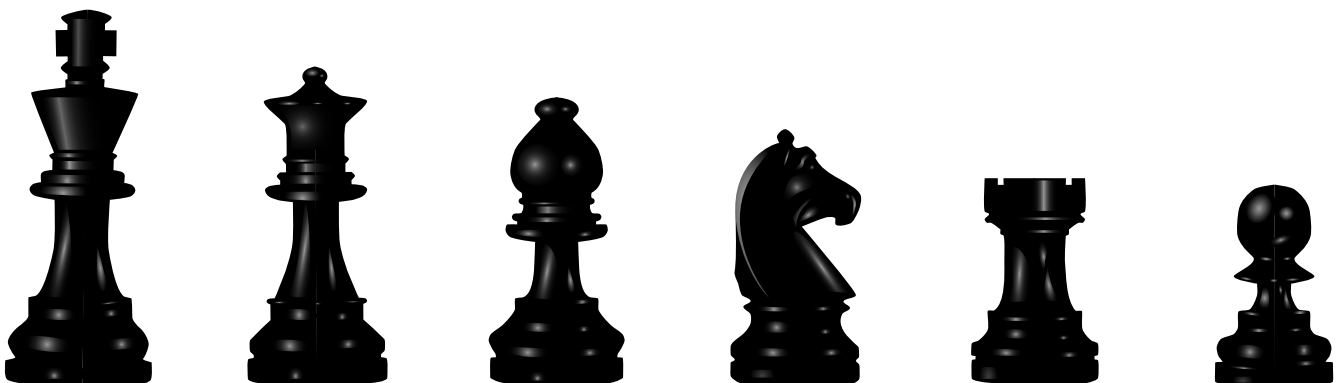
For example, a bank might realize 90 percent of its IT costs are related to maintenance and use APR to determine what can be retired and/or combined to cut costs. Or, a bank might not be in a position to introduce new technology, so through APR it will look at what it already has and figure out how to make it "lean and mean."

APR is an activity that falls under the APM umbrella. You need APM to make good APR decisions. You need an APM governance and continuous improvement process that is ongoing and, through this, you might identify the need to undergo APR.

# Portfolio management challenges

Key challenges to effective APM include the following:

1. **Organizational readiness:** For example, does the bank have proper recordkeeping? Does it know which applications it has? Is there an inventory? Successful APM requires you to first know what you have.
2. **Availability of application information:** As previously discussed, a lack of documentation is a common challenge with legacy applications. A qualified APM provider will identify and analyze existing sources of information (e.g., source code, databases, business owner and IT manager knowledge) to obtain as much information as possible on each application.
3. **Poor project execution:** There are many reasons for this, such as poor project management, running out of money, failing to understand the architectural complexity involved, etc. Strong governance and project management are needed, as well as senior sponsorship. Senior leadership is an imperative to make staff changes and other tough decisions.
4. **Duplication not always easy to identify:** Duplication is not always a black and white issue. Even if two applications are performing the same functions, they might perform those functions in different ways (e.g., two payroll systems may perform the same functions, but in different ways based on local legal requirements).



# Portfolio management success factors

## METHODOLOGY

To address these challenges, a comprehensive, ongoing and governance-based APM approach is needed. CGI's APM methodology lays out the following key steps:

1. **Plan:** This involves selecting applications for review and identifying the right metrics. Before beginning, find a project champion in senior management and engage stakeholders from technology, business and operations.
2. **Measure:** In this phase, a measurement plan is developed, rating scales are assigned to each metric (based on how well they reach target values), measurements are collected (using a web-based tool), and an application portfolio report is generated.
3. **Analyze:** At this stage, application issues and risks are identified (e.g., duplicate functionality) and a mitigation strategy is developed. Each application is assigned a risk score and plotted into a matrix based on business value and risk.
4. **Transform:** The overall "application portfolio risk versus business value" matrix developed in the previous stage determines which applications require transformation. Once these applications are identified, metrics developed for each will determine what kind of transformation is possible and most beneficial.
5. **Communicate:** This involves communicating the findings of the overall APM assessment to the right people. Typically, an application portfolio report consisting of assessment results and application scorecards is distributed.

## FRAMEWORK

An APM framework (APMF) is the foundation for effective APM. Such a framework should include a solid methodology combined with a powerful software toolkit. It should ensure the right people are involved and the right information is collected. It should also provide the following:

- **Scalability:** To address small, medium, large and massive portfolios
- **Flexibility and adaptability:** To meet the needs of different industries and markets
- **Transparency:** To ensure accountability and to support ongoing audits
- **Simplicity:** To ensure streamlined information collection, analysis and reporting

A strong APMF will support a number of key APM services, including the following:

- **Application inventory:** Inventory of applications across the organization
- **Risk landscape assessment:** Review and assessment of risks that threaten sustainability of the application portfolio
- **Application metrics development:** Determines which metrics are relevant and easy to collect
- **Application portfolio assessment:** Comprehensive assessment of the application portfolio
- **Application rationalization:** Focuses on reducing the overall application footprint and operating costs
- **Application transformation roadmap:** A roadmap for transforming the application portfolio
- **Application portfolio management office:** To provide ongoing governance and project management

In terms of supporting tools, the ideal APMF toolkit should automate information gathering, portfolio analysis and reporting. It should be flexible and configurable to adapt to varying client and project requirements. Overall, it should lower the cost, effort and risk for delivering APM services and ensure consistent, high quality client deliverables.

## Conclusion

The rise in digital consumerism, the emergence of new competitors, the pressure to launch new products and services, increasing regulations and the ongoing drive to save costs are leading banks to rethink their legacy landscapes and how to revamp their legacy applications to support future business growth. Both application modernization and application portfolio management (APM) are proven approaches to preserving and getting the most value possible from your legacy investments.

It is important to find the right partner to succeed at either or both. You need a partner with vast experience in delivering modernization and APM projects, proven frameworks, methodologies and tools, partnerships with leading solution providers, strong modernization and APM experts and capabilities, and a global delivery model.

CGI offers all of the above, and we welcome the opportunity to discuss how your business can benefit from modernization and/or APM. Visit our website, [www.cgi.com](http://www.cgi.com), or contact us at [banking.solutions@cgi.com](mailto:banking.solutions@cgi.com)





www.cgi.com

## cgi.com

---

With 68,000 professionals operating in 400 offices in 40 countries, CGI fosters local accountability for client success while bringing global delivery capabilities to clients' front doors. Founded in 1976, CGI applies a disciplined delivery approach that has achieved an industry-leading track record of on-time, on-budget projects. Our high-quality business consulting, systems integration and outsourcing services help clients leverage current investments while adopting new technology and business strategies that achieve top and bottom line results.

---

© 2014 CGI GROUP INC.

All rights reserved. This document is protected by international copyright law and may not be reprinted, reproduced, copied or utilized in whole or in part by any means including electronic, mechanical, or other means without the prior written consent of CGI.

---