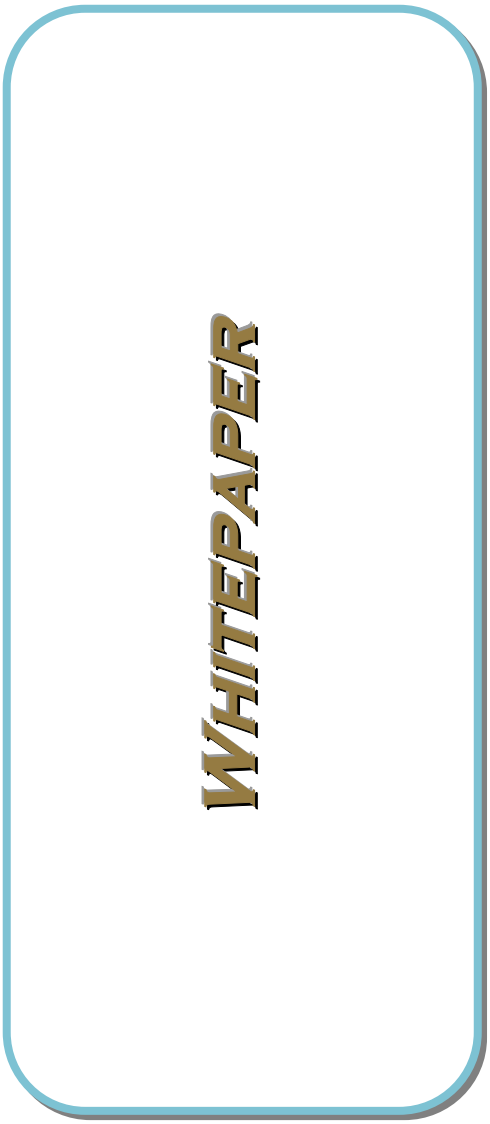
SYSTEMS TECHNOLOGY GROUP, INC.

INNOVATION, SERVICE, EXPERIENCE

LEGACY MODERNIZATION

- Bob Jonson



Abstract

According to industry estimates, COBOL remains the most widely deployed programming language running business-critical applications on legacy mainframe systems, accounting for nearly 75% of all enterprise transactions. CIOs and IT Managers handling mainframe based applications constantly face challenges such as:

- High costs associated with maintaining and updating applications on mainframe and legacy platforms
- Unmitigated risks related to running business critical applications on no longer unsupported hardware/software platforms
- Complicated development lifecycle due to lack of productivity tools, with higher time-to-market and difficulty in meeting regulatory compliance needs
- Inflexible and closed architecture systems that hinder web-enabling and integration with modern platforms
- Limited options for hardware and software
- Issues related to lack of skilled resources on legacy technologies, which are no longer included in academic curriculums
- Higher SLA time for customer due to a lack of real-time architecture options (i.e. batch systems)

This paper discusses “Legacy Modernization:” as a viable and safe solution and highlights comprehensive insights into STG’s credentials in the Legacy Modernization space. It covers various risk mitigation techniques along with their pros and cons and explores the reasons why organizations prefer a suite of modernization techniques over a single technique.

Myths about Legacy Systems

Legacy systems, like mainframes, which have evolved over a period of more than four decades, are well known in the IT industry for the Reliability-Availability-Scalability (RAS) factor. These parameters are fine if the business model doesn't change for 10-15 years or there are no new business needs and demands. This is impractical in today's dynamic and competitive business environment. Reliability of any enterprise platform will be much more secure, if the applications developed on these legacy systems are accessed from dumb terminals within the same network in a centralized architecture. Is reliability the same on contemporary platforms like Window-Intel (Wintel) for applications being accessed across different networks (Internet, Intranet, extranet etc), from different devices/ interfaces (desktop, laptop, mobile, handheld, etc.), by users having varied profiles (employers, customers, partners etc)? The costs associated with legacy systems, like mainframes, are per-transaction, so more transactions means more cost incurred. Consider a hypothetical scenario where the number of transactions needs to be increased on a legacy platform, based on a new competitive business initiative, without much increase in the company's financial bottom line. Is the legacy application truly scalable when IT costs exceed the income from new business initiatives? Mainframe systems are considered to be mammoth stores of centralized computing and processing power. But 'good enough computing' isn't adequate for mission-critical applications in a dynamic business environment. They need more-better scalability, better flexibility and better manageability. In recent years, Microsoft Windows and other open systems platforms have marginalized the mainframe computing value proposition. They are much simpler and clearer compared to mainframe systems, which are based on complex and perplexed capacity-based pricing models (MIPS per MSU).

Millions of dollars of investment and decades of development effort from generations of teams have ensured that legacy platforms support a wide gamut of business-critical functions and tasks. This makes legacy systems IT's biggest asset base. On the other hand, issues related to legacy systems, like aging of technologies and platforms, high operational costs, increasing scarcity of skills, and complex integration issues are key compulsions to modernize these legacy systems.

Legacy Crisis - Modernization is the way to overcome

Most organizations have shown little interest in investing in their legacy applications and platforms, beyond maintaining the same. The cost of this accounts for nearly 70-80% of the total IT budget, with only 20-30% of IT investments becoming available to cater new business demands and innovations. With most of the annual IT budget being dedicated to maintenance, there is little left for new developments and innovations. Decreasing the cost of maintenance of legacy systems is an obvious and appealing option, but is difficult to practice as many of the ongoing activities are supporting current business. If legacy maintenance activities continue to outpace new software development on contemporary platforms, eventually no resources will be left to develop new systems. Companies will be trapped in a "legacy crisis ".

Significant savings in operating and maintaining legacy systems releases funds and resources for new business initiatives and investing in innovative projects. Short-term savings are a first step to bridging the gap between business needs and IT functioning, resulting in alignment of IT systems with business. In recent times, innovation in the software industry has focused on the new applications for distributed, open systems platforms (like Microsoft Windows), using newer architecture and technology paradigms (like portal technologies, Web Services and Service-Oriented Architecture, SOA).

Long-term savings can be applied to funding innovation and gaining a competitive edge. So, "Legacy Modernization" is an attempt to evolve a legacy system, when conventional practices, such as maintenance and enhancement, can no longer achieve desired system properties.

Drivers for Legacy Modernization

Most legacy mainframe applications have existed for decades and need continuous enhancements and substantial maintenance to meet changing business needs. Monolithic mainframe applications are increasingly unable to keep pace with, and respond quickly to, dynamic business demands. An increasing number of companies want to upgrade their systems by leveraging existing IT assets (business rules, data and personnel), minimizing cost and reducing time.

In our experience, these are the main reasons why companies consider Legacy Modernization:

1. **High cost of maintenance**

Legacy systems hardware and software are very expensive. Pricing is non-competitive, due to lack of vendors. Original documentation is poor, as the applications were developed decades ago in monolithic programming paradigms. Subsequent maintenance and enhancement over the years, by generations of teams, have resulted in further complications in the application structure. Mainframe legacy applications need continuous maintenance, resulting in highly complex and unstructured source code. This exhausts valuable resources (time and money).

2. Lack of flexibility and integration

The monolithic architecture of legacy mainframe applications - where a single module or program contains logic of the presentation layer, business rules and data access logic - is a potential reason for inflexibility. A minor change in any layer (like increasing the length of a screen field or changing the structure of copybook) leads to major changes required in the code. Tight coupling of the business logic with the presentation layer adds to the complexity of integrating with new systems developed on contemporary platforms. Legacy applications, thus compound the difficulties in getting a set of applications to work together in an integrated environment.

3. Shrinking workforce and fading skill set

According to Gartner, "25% to 30% of employees with legacy skills will be eligible to retire in the next three years". There is an insignificant addition of programmers and systems administrators to the pool of people having mainframe skills. New talents are not being attracted to legacy mainframe development. The field has stagnated for the past 10-15 years, as all new advances are taking place in areas like .Net and Java. Even academicians are staying away. All these factors have resulted in fading mainframe skill sets. Sustainability of legacy mainframe applications will be jeopardized in next 5-10 years.

4. Higher time-to-market

Lack of frameworks, productive and advanced IDEs, debugging tools, and test automation makes development a tedious and cumbersome process in legacy environment. Organizations relying on legacy systems have a high time-to-market while responding to new business needs amendment in regulations and challenges from competitors. Even if new business functionalities are developed using modern technologies like .Net or Java, with highly productive Integrated Development Environments (IDEs) like Visual Studios or Eclipse, integrating these with the core business functionality running on legacy systems is a time-intensive task.

5. Risk of unsupported infrastructure

Many companies are running critical applications on legacy systems, using unsupported hardware or software. In the recent past, many large mainframe vendors have shut shop or stopped supporting their products/ platforms. The situation can be likened to a ticking time bomb, with a huge disaster just waiting to happen if major technical issues arise, which they will, sooner rather than later.

Legacy Modernization and migration of applications focus on increasing business competitiveness and IT flexibility, at a lower Total Cost of Ownership (TCO). Apart from improved Return on Investment (ROI), increasing business competitiveness means:

1. An increase in product and service innovation through shorter product lifecycles, and integrated product offerings

2. Strong customer intimacy through multi-channel access, rich and consistent information
3. aster business process integration resulting from M&A activities and business process re-engineering initiatives
4. Measurable business process efficiency, managed and optimized through enterprise and business unit dashboards and information analytics
5. Enhanced business user effectiveness through implementation of desired changes to existing IT systems, allowing implementation of SOA

The legacy issues described earlier in this paper (to contain or reduce mainframe costs, impending retirement of mainframe-skilled staff, detrimental impact on backlog, and business agility) are well known to most IT organizations. Their impact is increasingly being recognized in the CEO's office and the Boardroom.

Modernization track

STG's Legacy Modernization methodology consists of two broad phases:

1. Assessment - this phase consists of a number of steps

- Application Portfolio Analysis (APA), which involves business and IT stakeholder interviews to identify issues that need to be addressed in legacy systems
- Architecture Definition, which will recommend a future state architecture that is best suited to address IT portfolio modernization
- Modernization Strategy, which is the decision-making step that determines whether to keep, retire, replace or modernize applications
- Business Case Justification for the selected modernization strategy

2. Enactment - this phase, in the STG approach, covers governance, risk mitigation and planning. The planning takes into account all the aspects of modernization, i.e., modernization of applications, databases, infrastructure, and operations.

To embark on Legacy Modernization, organizations can start with an APA exercise. This helps in identifying the business value of the applications in the portfolio and the potential to increase it through modernization. APA aids strategic decision making:

- By addressing issues related to skill sets
- By helping assess the correct balance of investment in new priorities vs. maintenance of existing portfolio
- By providing optimum utilization of resources
- By obtaining insights, which form a critical input for defining the future architecture of the legacy systems
- Resulting in a better governance mechanism and scope for application portfolio rationalization

After APA, the first application for modernization should be one from the business functionality or Line of Business. Preferably, this would have high importance and low efficiency, so that reducing the cost per transaction adds substantially to the business bottom line. Assessment of the selected application(s) from technical and business process perspectives, by interviewing various stakeholders, will help identify pain points which need to be addressed by modernization. Insights from APA form a critical input for defining the future architecture of the legacy systems and addressing the pain points identified from assessment of the application. The future architecture of the modernized application can be defined by the process modeling tool to incorporate existing and new business requirements into functional requirements for an IT solution.

Business Case Justification estimates the TCO and expected ROI if a particular modernization strategy were to be implemented. The first imperative viewpoint from business case justification will be the operational cost comparisons between existing legacy systems and future applications on new infrastructure. This will help decide whether to go ahead with a particular modernization strategy or look for a better alternative.

Factors Determining Right Modernization

A number of factors, in combination, determine the appropriate Legacy Modernization techniques.

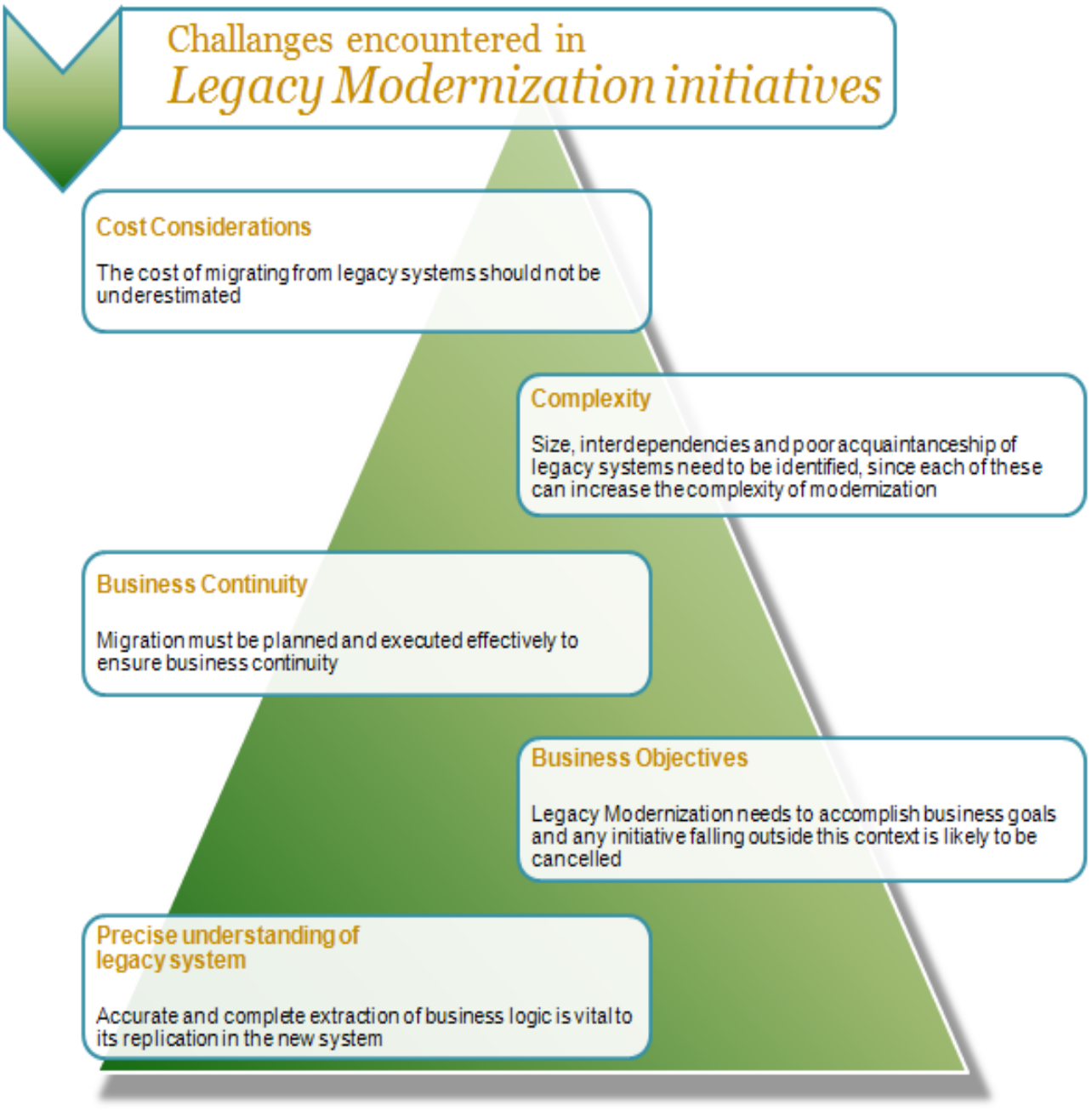
Cost of Migration: The total cost incurred (hardware, software and professional service charges) in migrating the legacy system using a given technique.

Effort for Migration: The total effort involved (internal, by the client, and external, by the professional services provider) in migrating the legacy system, using a given technique.

Elapsed Time for Migration: The time taken from the beginning of the migration program to the implementation of the migrated applications using a given technique.

Legacy Investment Reuse: The extent to which existing legacy code can be re-used in the modernized future application.

User Experience: The user experience of the modernized future application in comparison to the existing legacy application.



Business Process Change: The changes required to optimize the supporting business process of the application considered for modernization.

Skill Set: The skills needed to support and enhance the application in future, after modernizing the legacy application. This is also a crucial factor for executing modernization.

Architecture and Business Functionality: The comparison between the legacy system and the modernized application with respect to these two parameters

Dependency on Legacy Platform: This factor determines if any dependency with legacy systems will exist post-modernization.

Risk Mitigation: Determined by the above factors, taken along with migration type (invasive or non-invasive).

Usually, for a non-invasive migration type, the risk mitigation plan is not of high priority compared to an invasive migration type.

Legacy Modernization Techniques

The challenges associated with legacy mainframe systems, pointed out earlier in this paper, are real business hurdles for most organizations looking to leap to the next level of growth in an extremely competitive flat world. These issues include high TCO of applications on legacy platforms, a shrinking labor pool that possesses the required legacy technology skills, information lock-out in closed architecture legacy platforms, and higher time-to market for new products and offerings. STG's Legacy Modernization strategies address them with the following adaptive techniques:

1. Integration via service-oriented approach

Legacy systems have valuable information in terms of the business logic and data locked in their closed architecture environment. In today's distributed paradigm, organizations are faced with the huge task of unlocking this valuable information and serving it in a seamless manner to other distributed IT systems in the enterprise IT landscape, enabling each system to benefit from the assets of the other. Integration with legacy systems in a service-oriented approach can help enterprises address issues of information lockout.

Integration of legacy mainframe systems with other, newer, distributed systems in a service-oriented approach is the process of exposing business logic and data embedded in mainframe programs as well-defined, reusable services. This is achieved by designing services based on SOA principles and integrating them in any of the legacy application layers. So integration techniques can be broadly categorized into the following types:

a) Presentation Integration Technique (popularly known Screen Scraping)

b) Programmatic (Business Logic) Integration Technique

c) Data Integration Technique

Integration is a useful way to leverage existing investments in legacy systems with minimum effort, thereby extending the life of the legacy assets.

2. Re-hosting:

Re-hosting is a non-invasive technique for migrating a legacy system to a new hardware platform. It is possible to achieve execution of mainframe applications within a new environment with a high degree of automation (80%).Typically, re-hosting addresses the higher costs of maintenance of outdated platforms. However, it does not address the challenges related to application code. The process of re-hosting applications involves shifting system inventory from mainframes to new platforms, building specific references of applications (removing proprietary language dependencies), re-compiling, and executing on the new platforms.

In re-hosting, the business logic and data of legacy mainframe applications remain intact in the new platform. Re-hosting is often necessitated by the increasing power of contemporary hardware platforms and the steady increase in the maintenance cost of legacy systems. Re-hosting to

newer hardware platforms, like Wintel, often leads to a sharp reduction in maintenance and operational costs.

3. Re-engineering:

Re-engineering is the big-bang approach to modernizing systems. It works by gathering requirements from existing legacy applications and redeveloping them on newer platforms using new technologies. Re-engineering seeks to understand existing systems from a process and code perspective, designs viable models and builds new applications on that basis. It is the ideal way to capture hidden organizational knowledge and migrate it to new platforms. By leveraging the target platform, re-engineering is a means to revitalize business competitiveness.

Re-engineering is a two-phase approach. The first phase comprises reverse engineering design and code artifacts using tools. The second phase comprises forward engineering to a new platform and architecture, incorporating new business functionality and requirements. Although re-engineering a portfolio is suitable for high business value applications and assures lower TCO over longer periods, complete re-engineering leads to higher elapsed time and entails a higher risk migration plan. Adoption of modern technology with new architectural paradigms like SOA, through re-engineering, leads to increased agility and flexibility, and reduces operational expenses. This is achieved by moving applications to a more commodity-based and cost-effective operating environment, like Wintel.

4. Package Implementation:

This approach implies abandoning legacy applications and replacing them with commercial-off-the-shelf packages. This option focuses on building a portfolio with the best packages and components available from third-party vendors. However, reuse of existing business logic is not possible with this approach. Replacing legacy applications with commercial packages can be quite challenging as the commercial packages offer standard domain business processes that differ from the homegrown custom build application. This often means undergoing some level of business process re-engineering or customizing packages and rewriting business logic. Both of these can be time-consuming and expensive.

Replacing legacy applications with commercial packages drastically reduces maintenance and enhancement on the source code, carried out by the internal IT team. The onus is on the package vendor to fix production bugs and develop and implement new functional enhancements.

The first technique (integration) is the co-existence of legacy and contemporary platforms, as the applications in the legacy environment are retained. All the other techniques are about completely transforming legacy applications to a more cost-effective, flexible, open platform, like Windows. Being adaptive (reactive) in nature, these techniques are considered when organizations are struggling with legacy systems. Apart from the above reactive approaches to tackle issues related to legacy platforms, STG also proposes a preventive approach - that of Legacy Documentation. If the current cost structure and operational efficiency are not really daunting, or the retirement of skilled legacy resources is not imminent, Legacy Documentation can be considered to decipher legacy systems that are not properly documented.

Legacy Documentation is a non-transformational modernization technique to capture the business value embedded in the legacy systems code. So, an organization incurring high costs in carrying out maintenance with an internal IT team can look for an another option with "Transition-to-Transformation" approach. Legacy Documentation will be more effective if it is done alongside maintenance activities, so that during the transition phase itself, legacy systems can be documented. Legacy Documentation increases the readiness of the application for the transformation phase, which can be carried out any time after successful completion of the transition phase. Legacy Documentation also addresses regulatory compliance and increases the productivity of maintenance and enhancement activities.

Technique / Parameter	Re-hosting	Re-engineering	Integration	Package Implementation
Cost	Low	High	Moderate	High
Effort and Time	Low to Moderate	High	Moderate	High
Reuse of Existing Investment	High		Moderate to High	Very Low to None
Risk	Low	High	Moderate	High
User Experience	Same As-Is	New and Rich	New	New
Business Functionality	Same As-Is	As-Is and Newer	As-Is	Less
Agility	Same As-Is	High	Low	Moderate
Skill set	Same		Mixed	Newer
Business Process Change	No	Yes	No	Yes
Dependency on legacy systems	No	None	Exists	None

CONCLUSION

Legacy Modernization requires detailed planning and program management. The effort involved in shifting workloads out of mainframe legacy systems should not be underestimated. Businesses have a need to enlist the services of a solution provider with expertise in consulting, technology and management.

Re-hosting and Integration options focus on managing existing portfolios better, to contain mounting operational expenses and increase applicability through data unlocking. These typically take less time and effort to implement, and make maximum reuse of the existing investments in mainframes. Re-engineering and Package Implementation options focus on retiring aging legacy applications, and transforming the portfolio by building with an eye to the future. While they provide benefits in controlling operational costs, they also provide an architectural overhaul to the portfolio, increasing its life and providing a foundation for building future-state architecture.

Depending on the objectives of the organization, different options need to be chosen. The key parameters that affect this decision need to be ploughed into a business case and the ROI needs to be measured.

With evolving contemporary non-mainframe systems delivering mainframe levels of performance, reliability, security, and scalability, organizations can look to reduce infrastructure and transaction costs by migrating to cost-effective commodity-based platforms, like Wintel, with a lower total cost of ownership (TCO). Agility of the modernized systems will increase competitiveness through new innovations and offerings (product or services). These can, in turn, augment market-share.

STG's Legacy Modernization solution is part of on organic IT initiative, an internally led business transformation program initiated by its founders, based on their decades of experience in the mainframe as well as SOA domain. We have built our own frameworks and use tools in APA, Business Process Engineering (BPE), and business case justification to achieve a rapid and complete assessment phase. From the numerous Legacy Modernization projects we have executed for various clients, we have developed matured methodologies and workbenches for faster execution of modernization projects with risk mitigation.

About the Author:

Bob Jonson is the Chief Technology Officer at Systems Technology Group, with rich experience in various development and enhancement projects on legacy platforms and SOA architectures. He has also led and assisted Legacy Modernization projects. Bob and his team are involved in the development of sevral modernization projects for legacy systems on the Microsoft platform.

Bob's current area of work includes Legacy Database Modernization, and Collaborative Reverse Engineering, SOA Integration, and Process-enabling Legacy Systems,

Systems Technology Group defines, designs and delivers IT-enabled business solutions that help small and large companies as well as government agencies succeed. These solutions focus on providing strategic differentiation and operational superiority to clients.

STG creates these solutions for its clients by leveraging its IT and business expertise along with a complete range of services.

With STG, clients are assured of a transparent business partner, world-class processes, speed of execution and the power to stretch their IT budget.