

WHITE PAPER: LEGACY MODERNIZATION

# An Automated Approach to Legacy Modernization

JANUARY 2010

Mike Helft and Cindy Peake

CA MAINFRAME SOLUTIONS

---

# Table of Contents

---

## Executive Summary

---

SECTION 1 2

### What is Legacy Modernization?

Application Portfolio Assessment

---

SECTION 2 3

### Why Modernize?

---

SECTION 3 4

### Alternative Modernization Approaches

---

SECTION 4 7

### CA Legacy Renewal Approach

Automated Business Rule Extraction and Documentation

Legacy Data and Logic Interpretation

Deliver Modernized Solution

What is Model Driven Development?

---

SECTION 5 10

### Benefits of CA's Legacy Renewal Approach

Benefits of Automated Extraction and Conversion

Benefits of Model Driven Modernization

SOA Modernization

---

SECTION 6 13

### Conclusion

---

## ABOUT CA

### Back Cover

# Executive Summary

## Challenge

---

Today's businesses demand new and improved software to enable business growth and the ability to rapidly develop and deploy new products. At the same time, it organizations face continued pressure to speed application delivery while reducing costs. As if that's not enough, technology change is happening at a rate faster than ever before and organizations are continually challenged to assess these new technologies and then implement those technologies that will benefit their business. This creates the ongoing challenge of balancing investments in the existing software infrastructures with investments in future technology opportunities. Legacy modernization is one solution to this challenge.

## Opportunity

---

Legacy modernization is about retaining and extending the value of legacy applications, reusing the intellectual property buried in these systems and transforming them to more modern architectures. There are numerous drivers that compel an organization to explore legacy modernization, as well as numerous approaches to accomplish this task. Each approach has various pros/cons and results. This paper explores these approaches and introduces a cost effective, low risk alternative that combines knowledge based services and automation tools to move your mission critical legacy applications into the 21st century.

## Benefits

---

CA Legacy Renewal Solution enables organizations to:

- Extract and reuse valuable business rules embedded in legacy code
- Document legacy business rules and processes during extraction
- Enhance discovered business rules to optimize process performance
- Modernize to new platforms and architectures such as J2EE, .NET, SOA
- Insulate modernized solutions from future technology changes
- Significantly reduce time, cost and risk associated with legacy modernization

## SECTION 1

# What is Legacy Modernization?

IT organizations face ongoing pressures to improve customer service, increase profits and provide more timely access to information. A key strategy to meet these demands is delivering continued use and value from existing legacy applications.

A legacy application may be defined as any production enabled software application, regardless of the platform it runs on, language it is written in, or length of time it has been in production. Legacy applications contain the business rules and data that make up the intellectual property required to run the business.

Legacy Modernization is about application use and reuse, building on the strengths of the past and combining them with the opportunities that present and future technologies offer. Legacy modernization is the conversion of aging applications to more modern architectures.

### Application Portfolio Assessment

Software applications are an asset to any organization. As enterprises evolve, so do their applications. The first step toward legacy modernization should be to assess the applications within your IT portfolio.

- How do you continue getting value from these applications?
- How do you integrate them with newer applications?
- How do you balance the decision to invest in the increased business opportunities of today's new technologies versus maintaining the investment in your portfolio of existing applications?

Options for evolving your application portfolio range from leaving the application "as is," integrating the application, migrating the application to new technology, or eliminating the application altogether. Balancing the choices is not easy and many factors need to be considered to determine what is best for your organization.

- How much business value does the legacy application currently provide?
- What are your current operation costs of the application (infrastructure, license fees, etc.)?
- Does the application meet scalability and performance requirements?
- Do you have skilled resources to maintain the application going forward?
- Are you able to quickly modify the application to adapt to changing business needs?
- How will business and technology trends affect your IT organization in the next five years? Beyond that?

## SECTION 2

# Why Modernize?

There are numerous drivers that compel an organization to explore Legacy Modernization strategies.

**APPLICATION INTEGRATION** One of the main drivers for organizations to consider Legacy Modernization is the need to integrate applications from diverse parts of the organization or across organization boundaries while maintaining the tight cost controls that have become an accepted fact of life. The need for integration is driven by many factors such as acquisitions or mergers, the need to streamline business processes or new business partnerships. There are two types of application integration that are important to organizations. One type of integration is where applications are tightly coupled together. And the more frequent integration, which can be better described as interoperability, is the requirement to get diverse applications working together. It is clear that for applications to interoperate, the organization must understand how these applications help the organization achieve its business goals. It is also accepted that older applications were created as highly complex and poorly architected solutions to narrow business problems. In today's age of cross organizational systems it is necessary to consider these problems from a different point of view. This is where the ability to extract and interpret business rules becomes critical.

**PLATFORM MIGRATION** Many organizations are defining strategies to move away from mainframe based solutions to distributed environments. In most cases, these distributed platforms, such as J2EE and .NET, offer a lower cost alternative, as well as a more flexible infrastructure. The need for agility using technologies such as Web services and Service Oriented Architectures (SOA) is also driving organizations to review their platform strategies. These migrations don't happen overnight and an evolution strategy that enables applications to co-exist in multiple environments is often needed.

**REGULATORY COMPLIANCE** Business processes are an organization's mechanism for creating and delivering value to its stakeholders. The ability to harvest business processes from legacy systems, document them and transform them into new modern solutions, assists organizations in meeting process compliancy requirements associated with legislated standards such as Sarbanes Oxley and HIPAA in the U.S. and BASEL II in Europe.

**COST REDUCTION** The need to reduce operation costs is a significant driver for many organizations to modernize their legacy systems. Many legacy environments, such as the mainframe, are expensive in terms of both hardware infrastructure as well as costly software license fees. There is an urgency to downsize to what are seen as more cost effective platforms. There is also significant cost reduction to be achieved by having applications that are in more modern development environments that enable increased developer productivity and reduced maintenance costs.

**COMPLEXITY REDUCTION** Overtime, many legacy systems have been modified so many times, by so many different developers that the code has become very complex and inflexible. These systems take forever to change and in many cases are incapable of being changed. Organizations need a way to move these "spaghetti code" legacy systems into an environment where they can be maintained quickly and easily.

**SPEED TO MARKET / FLEXIBILITY TO ADAPT** Today's businesses demand new and improved software to enable business growth and the ability to rapidly develop and deploy new products. Many enterprise systems in production today were written in the 1970s and 1980s and unless an organization was very lucky, the systems were poorly architected, poorly coded and are now impossible to maintain. This is not to say the developers of the day were inept or incompetent but rather that our understanding of application development has advanced significantly to enable us to develop adaptable and flexible solutions.

The requirement is now to move those old mission critical applications into the 21st century. This means there is a need to find a mechanism for extracting the business rules that are well hidden in the code. The ability to reuse components from legacy applications can prove to be of major competitive advantage.

**TECHNOLOGY CHANGE** Technology change is happening at a rate faster than ever before and organizations are continually challenged to assess these new technologies and then adapt their software applications to leverage those technologies that will benefit their business. Some organizations need to modernize their legacy applications to web infrastructures that will enable them to compete and provide improved customer service. Other organizations need to integrate their legacy applications using Web service technologies and Service Oriented Architectures. Organizations have invested significant amounts of time and money in the development of their core business systems. As technology gets more sophisticated and organizations grow more and more dependent on these core business systems, it is becoming crucial to ensure that this software is adaptable and maintainable.

**SKILLS AND BUSINESS KNOWLEDGE** A skills shortage is being widely anticipated as we move further into the 21st century. Organizations have been dependent on their core mainframe programmers to understand and maintain the applications that have been in production for many years. In most cases these applications have been patched time and time again by different programmers for different reasons. This means businesses have applications of considerable complexity, probably undocumented and with no single person understanding the rules embedded in the code. Therefore many organizations need a way to migrate these legacy systems, which were hand written in mainframe languages such as COBOL and Natural, into an environment that can be understood and maintained by non-mainframe skilled system analysts and developers.

### SECTION 3

## Alternative Modernization Approaches

There are several approaches and methods available to organizations needing to modernize their legacy applications. Each approach has various pros and cons and numerous factors should be considered when choosing a modernization approach:

- What developer skills currently exist in your organization?
- How quickly do you need the modernized solution?
- Does the legacy system contain business information that provides your organization competitive advantage?
- What platform(s) do you want your modernized solution to run on? Could this change in the future?

- What are the integration needs of your modernized solution?
- Does your legacy system support your current business practices?
- Are your current business processes optimized?
- How agile does the modernized solution need to be?
- How often do you need to implement process changes? Legislation changes? Technology changes?

Several of the commonly used modernization approaches are discussed below.

**MANUAL HAND-CODING** In this approach the code is manually transformed from mainframe based applications to “other” platforms. This approach delivers a customized solution that can be built exactly to suit. However, using hand coding techniques to rewrite the legacy system from scratch is an expensive and long process which has numerous risks attached to it:

- The code is capable of being interpreted in different ways by different staff involved in the project.
- These projects typically have long delivery cycles. Estimates have shown that manual conversion projects often impose unacceptable periods when business changes need to be frozen. If the application being converted is dynamic, meaning business changes are still being implemented, then the period for conversion has to be very short — this is unlikely in a manual rewrite project.
- The system may be as much as 25 or 30 years old and has been maintained by several programmers. It likely has little or no documentation and frequently no definition of business rules.

**WRAPPING** This technique uses existing application code, but allows modernization of the look and feel and integration with other applications via code written for this purpose. This approach is useful if all that is required is a temporary solution while a full migration effort is planned. It allows for rapid deployment to the web and the ability to quickly integrate diverse applications. However, there are some issues with this approach:

- With wrapping techniques, the legacy code is still out there. This doesn’t address the ongoing “spaghetti code” maintenance issues and declining skill set issues.
- Wrappers often offer simple connectivity to an API, but include no real intelligence about the business level interface — they typically don’t provide a better understanding of the business rules buried in the legacy code.
- Wrappers are sometimes a good approach for portions of an application that are static and not likely to change often. However, wrappers may not be the best approach for applications that are dynamic, which will likely require ongoing modifications due to changing legislation, process improvement, business growth, etc.
- Wrappers are often built to a specific version of an application, which may be different from your installation and may not have the ability to “sense” versions and make adjustments.

**PACKAGES** Many legacy systems were developed before package applications were available on the market, so in some instances, modernizing a legacy system by purchasing a package is an alternative approach. However, in some cases, the business rules that are embedded in the legacy system provide competitive advantage in an organization and can't be replaced by a software package. Another consideration when choosing this option is how much effort will be required to modify the package so that it maps to the organizations business processes. You don't want to make the mistake of modifying your business processes to suite the package application. In addition, when replacing legacy systems with package applications, organizations must consider the impact of losing control of managing and maintaining the code.

**AUTOMATED CODE TO CODE CONVERSION** There are point solutions in the market that provide the ability to programmatically move code away from "old" languages to newer, more modern languages. Although these tools are only applicable to specific scenarios and only handle part of the transformation process, there are some benefits of this approach:

- It enables the organization to guarantee it is retaining its investment. If a business rule is embedded in a piece of COBOL or Natural code, it will become embedded in a new piece of Java.
- The newer language will allow the organization to take advantage of more modern application architectures. Code moved to Java can be easily deployed to the web.
- Access to newer languages and application architectures means the organization can take advantage of new features that are available in these environments but not in their traditional arena.
- Because there is a potentially larger pool of developers and the application has been migrated away from the mainframe, maintenance costs can be reduced.

These benefits may provide value to organizations that need to rapidly move applications to a more modern environment. However, there are some core issues that should be carefully considered when looking at this approach.

- This approach converts code to code and it makes no attempt to understand the business rules that are embedded in that code. The organization is no better in terms of the business benefit it can derive from its applications — the code it did not understand before, it still does not understand, it's just now in a more modern language. Adaptability and reuse are as hard in this scenario as they were before the conversion took place.
- The ability to take advantage of the features in the new language is not as easy as might be portrayed. For example if the conversion has been to Java then it will require major re-architecting of the application to identify and deploy Enterprise Java Beans (EJBs). So, if technologies such as EJBs, Web Services or SOAs are part of the modernization goals, then simple code to code conversion will not meet the organizations requirements.
- The application that has been converted it still a legacy application. If the language that it has been converted to has new features and facilities added to it, or if in the future it becomes unsupported, unfavorable, or has scarce development resources, then the organization will find itself in exactly the same situation as it is today.

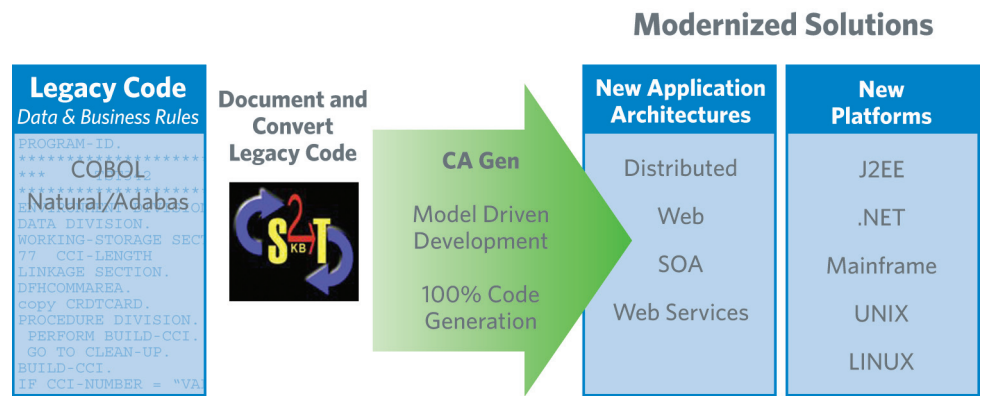
## CA Legacy Renewal Approach

Most organizations fully understand that their legacy systems contain not only useful, but vital business information and business rules. But the task of extracting this value from these legacy systems has remained difficult and time consuming and therefore many organizations have simply avoided this “daunting” task. The key to overcoming these barriers is to automate as many of the tasks associated with legacy modernization as possible. Substantial automation of the process makes legacy transformation an economically attractive and low risk proposition compared to other modernization options such as rewriting or replacing the legacy application. Automation, combined with a repeatable methodology will simplify the tasks as well as speed the process of modernizing legacy solutions.

FIGURE A

Highly Automated Conversion of Legacy Code to Model-driven Environment, Enabling Modernization into New Technology Solutions.

### AUTOMATED LEGACY MODERNIZATION



The Legacy Renewal Solution from CA uses automated extraction and a model driven approach to successfully transform legacy COBOL or Natural code to new solutions that implement modern technologies and architectures such as J2EE, .NET, Web services and SOA.

### Automated Business Rule Extraction and Documentation

A key step in successfully modernizing applications is to analyze the existing legacy code. CA uses Evolveware’s S2T Technology to automate the extraction of business rules, data and workflows from legacy COBOL or Natural/ Adabas code. The extraction process uses an iterative approach to maximize the automation — on average, as much as 70-85% automated conversion of legacy data and logic. Thorough reports are created to assist in manually converting the remaining legacy code.

There is often an assumption that all code should be converted, however this is typically an incorrect assumption, because some of the legacy code may be irrelevant to the modernized solution. During the extraction process, reports are generated that enable the organization to better understand unknown or poorly documented processes, as well as simplify complex logic constructs that have likely been developed over several years and by several developers.

### Legacy Data and Logic Interpretation

One of the biggest challenges in a migration exercise is to translate potentially meaningless code into well understood business terms. A key to success of a legacy migration project is to rename and reorganize the legacy data components that have been identified, very early in the process. If the data interpretation is done upfront, then the translation of the actual logic can focus on a better understood data structure, which can significantly speed the process. Similarly, logic interpretation involves translating meaningless COBOL statements into terms that better describe the business processes and can be understood by all members of the team. This translation can also be accomplished in the model driven development environment, however, experience has shown that intercepting the code early on during the conversion process is more efficient and speeds the overall conversion process.

### Legacy Conversion Output

Results of the legacy conversion process include two key deliverables:

- **A BUSINESS MODEL** This contains all the elements that have been extracted from the legacy source code and translated into objects that are understood by the modeling environment. It is fully documented so it is possible to trace each object in the model to the originating piece of code.
- **DOCUMENTATION** Equally important is the documentation, which summarizes what has been converted and just as vital, what has not been converted. This document has proved very useful during the forward engineering phase of the project.

### Deliver Modernized Solution

Once the business rules and data have been successfully extracted from the legacy system, they are automatically converted into a business model. From that point you can use a model driven approach to enhance the business rules to reflect current business practices and then forward engineer those processes into new application architectures and modern technologies. CA's premier model driven development environment, CA Gen (formerly known as AllFusion® Gen), provides code generators that enable organizations to deploy their new applications to numerous modern platforms and architectures such as J2EE, .NET, UNIX, Linux, Web Services and SOA. And for organizations who wish to remain on the mainframe, the robust code generators in CA Gen enable organizations to modernize their applications without getting off the mainframe. This flexibility helps reduce the risks of modernization by providing platform independence, where systems are 100% generated for multiple environments from a single model, reducing risks by providing an evolutionary path towards modernization.

### Data Conversion

After the data and process models are created, the new application logic and databases are generated. Another significant challenge associated with application modernization is data migration. The Legacy Renewal Solution offers flexible data modernization options which can leverage existing database environments, or transition to new ones, or a combination of both. If data migration is required, CA uses data transfer technology from partners such as Treehouse Software, Inc. to facilitate conversion of data from one database environment to another.

## What is Model Driven Development?

Model driven development uses models to visualize and manage the architecting and development of business applications. The purpose is to extract the complexity of the business into a set of predefined diagrams that enable organizations to separate business processes from technology implementation.

**INTEGRATED MODELING ENVIRONMENT** Modeling diagrams illustrate the major pieces of a system such as data requirements, business rules, user interfaces and process flows. With an integrated model based environment, the modeling objects which make up these diagrams are stored in a central repository, where they can be shared by the entire team throughout the development process. Robust security protects the integrity of the models objects, enabling parallel team development and reuse. When changes to the system need to be made, instead of sifting through and modifying a bunch of system code, changes are made directly in the model. And because it's an integrated modeling environment, this means when an object is changed, it is reflected throughout the entire model. By extracting legacy business logic and converting that logic into meaningful process and data models, organizations can gain a better understanding of their business, apply process improvements and maintain the new system much more efficiently.

**PLATFORM INDEPENDENT DEVELOPMENT** Model based development enables developers to build and design their applications independent of the deployment environment. From a single business model, they can deploy systems to their platform, database, middleware and language of choice. This provides organizations with a modernization approach that enables migration to multiple platforms NOW and in the FUTURE. Many organizations do not know what platforms they are going to target five years from now. CA Gen enables organizations to modernize to J2EE and .NET frameworks as well as mainframe solutions — or migrate slowly over time, with a co-existence strategy in the meantime.

**TECHNOLOGY INSULATION** Over the last two decades, application development technologies have changed from 3270 green screen and batch architectures, to distributed client/server computing, to web and wireless infra-structures, J2EE, .NET and now Web services and SOAs. The rate of this technology change has accelerated considerably over the last decade with no signs of slowing down. An organizations ability to adopt new technology and leverage it to achieve competitive advantage is a key to survival in today's fast paced markets. Model driven development, combined with 100% code generation, enables organizations to develop systems that leverage latest and greatest technologies, without having to understand all their underlying details. Organizations can regenerate their models to incorporate new technologies without having to spend the time and money to completely rewrite their systems.

## Benefits of CA's Legacy Renewal Approach

Organizations continually have to weigh the benefits of short term speed, with long term productivity and performance. CA's Legacy Renewal Solution — which uses automated extraction and conversion, model driven development and automated code generation — provides the best of both worlds, speed along with long term efficiencies such as reuse, maintenance productivity as well as platform and technology independence.

### Benefits of Automated Extraction and Conversion

Automation of the code extraction process significantly reduces the time and cost of identifying and documenting valuable business rules locked inside legacy code, enabling organizations to reuse their own intellectual property. Automating the conversion process also reduces the risks of legacy code conversion by minimizing errors and reducing the resources needed for the modernization effort. The methodology defined by CA Technology Services™ uses an assessment process that takes a sampling of code through the conversion process to estimate the amount of code that can be converted automatically. The level of automation depends on many factors such as the “condition” of the legacy code — whether it is monolithic or structured in nature and whether its batch or transactional. This assessment process helps IT executives make informed decisions, also reducing the risk of the overall project.

### Benefits of Model Driven Modernization

There are numerous benefits of using a model driven approach to modernize legacy systems. CA's Legacy Renewal Solution enables organizations to:

#### Increase Focus on the Business

When models are used to represent the business problem, it is easier to see what has been missed, what has been misunderstood and what has been needlessly included. When model based development is combined with 100% automated code generation, as with CA Gen, developers are able to focus on analyzing and improving the business, instead of worrying about tedious programming issues such as referential integrity, middleware interfaces, error trapping, HTML formatting and the list could go on and on. The use of models also improves communication between IT and the business. Modeling diagrams that are drawn using a clearly defined set of rules and syntax ensure that those who read the diagrams communicate effectively and precisely.

#### Significantly Increase Developer Productivity

As discussed earlier an integrated modeling environment provides significant increases in both development and maintenance productivity.

- **100% AUTOMATED CODE GENERATION** The consistent modeling syntax used in a model driven environment enables code generators to be used to create the system code. 100% automated code generation is not a myth, nor is it new. CA Gen has been generating 100% system code for its customers for over twenty years. This means your entire modernized solution is automatically generated, from the application logic and communications infrastructure to the web servers and browser interfaces. No manual coding is required. This enables organizations to speed time to market for competitive opportunities, react more quickly to change and reduce overall maintenance costs.

- **REUSE STRATEGIES** To fully maximize application development productivity and responsiveness, organizations need to focus on improving levels of reuse. A model driven development environment, like CA Gen, provides the ability to create reusable services and components as well as the mechanisms to understand and share them. Many organizations understand the benefit of and are trying to implement, a reuse strategy. This requires a significant organizational shift and using a model driven approach to modernize legacy systems can help ease this transition to a reuse culture.

### **Ensure System Quality and Performance**

Most applications that organizations are modernizing are mission critical solutions that run their business. Model driven development helps ensure quality and consistency of these solutions through reuse. Components and services that have already been defined (and even tested) are known to be of high quality and are used consistently by all development teams. The proven code generators in CA Gen create applications that are “native” to the platform they run on, optimizing system performance. Over time, the transaction throughput or number of system users often increases and the application needs to scale to accommodate this growth. Using a proven model driven development environment with code generation, the system can be quickly and easily regenerated from Windows to UNIX to Mainframe to Web — the application can scale without worry.

### **React Quickly to Change**

Whether it's a business change, platform change or technology change, model driven development enables organizations to implement that change more quickly and with fewer resources. User requests, legislative requirements and process improvements are made to the model, not in the code, so developers are able to react quicker and fewer resources are needed to maintain the system. Platform independence and technology insulation enabled by model driven development provide an organization with significant flexibility. For example, modernization does not have to mean getting OFF the Mainframe. CA Legacy Renewal enables organizations to generate batch, 3270, distributed and web solutions for the Mainframe. If desired, these modernized Mainframe solutions can then leverage J2EE, .NET and/or Web services as the organization is ready — without rewriting any code.

### **Maximize Investment and Minimize Risk**

Modernizing a legacy system can be a high risk proposition, not to mention, very costly. Using a phased approach methodology, in combination with proven technology, can significantly reduce these risks. An important risk to be managed is ensuring that the quality and performance of the newly modernized solution is consistent with expectations. The platform independence provided by the CA Gen model driven environment significantly reduces this risk.

Since the code is 100% generated from the business model, significant time and resources are not spent on coding the system for, say Java and EJBs, only to find out it doesn't perform as expected on a J2EE application server. With CA Gen the system can be generated for Java and EJBs and then if it doesn't perform as expected either specific transactions or the entire system can be regenerated for another platform, such as Mainframe, UNIX or .NET. Platform independence also insures that your current and future hardware and software investments are protected.

- **STAFF AND SKILL MIGRATION** There are also major costs and risks associated with migrating developer skills to a new language or technology. Using a model driven environment with code generation significantly reduces those costs and risks because developers are able to be productive much more quickly. With CA's Legacy Renewal Solution, in depth knowledge of platforms, databases, middleware and languages is not required to modernize to them. Once they understand how to use the model driven environment, developers are instantly productive when moving between projects that may be using completely different platforms or technologies. Organizations wishing to move toward a Service Oriented Architecture using Web services can focus on implementing a reuse strategy, rather than worrying about learning all the underlying technologies such as XML and SOAP.

### SOA Modernization

Many organizations wish to transition to a Service Oriented Architecture as part of their modernization efforts. This is not something that can happen overnight. Some of the biggest efforts when transitioning to an SOA are not related to technology issues, but rather organizational ones. So, evolving from a 3270 Mainframe Architecture to a more agile Service Oriented Architecture is best handled in phases. Organizations should consider modernizing their legacy applications to a distributed web architecture initially, providing significant process improvement and value to the business. At the same time, longer term strategic plans can be put in place to transition to a full agile SOA. These plans need to address numerous facets such as organization change, system change, process change and more. Model driven development provides the foundation for the critical first steps in modernizing to an SOA.

To maximize the return on investment of application modernization, business units and IT organizations must move their vision beyond single applications and look toward the higher level business processes. Instead of focusing on modernizing and developing single applications, organizations need to consider the entire set of services that, taken together as a process sequence, represents a higher level business process flow. Implementing SOA, using a model driven approach, can usher in a new way of thinking about building and maintaining ROI within business applications. Integrating services with automated work flows provides a higher level view of Business Process Management (BPM) across multiple applications. Combining model driven development, SOA and BPM during application modernization will help reduce and eventually eliminate, the next generation of legacy applications. Within an SOA, entire applications will no longer require modernization. Applications can be redesigned, re-platformed, redistributed and integrated by modifying the individual services that comprise the application.

Using a model driven approach can significantly increase the speed at which you can assemble and re-use existing processes to deliver services. By providing short term business benefit, while at the same time implementing long-term strategic value, this phased approach to architecture modernization has proved much more realistic and successful than the "big bang" approach.

## Conclusion

Most organizations realize that at some level, they must modernize their applications to remain competitive. Whether that means doing business via the Internet, implementing a more flexible application infrastructure, or simply better integrating their systems, organizations need to review their application portfolio and assess new technology opportunities. CA's Legacy Renewal Solution offers platform independent, Service Oriented Architecture application development that enables businesses to integrate, modernize and extend their legacy systems in a productive and future proof way. By introducing significant automation for the migration process, along with the benefits of a model driven code generation capability, the CA solution can drastically reduce the time and effort necessary to recover the business value that is buried in legacy applications.

The business will achieve further return on investment from its "old" applications by extracting the business rules and being able to reuse them in new systems that may, or may not, make use of new technology. Because the applications are self documenting through the model it becomes significantly easier to demonstrate compliance to legal and auditing regulations.

The model driven approach, irrespective of the source of the model, allows businesses to achieve faster time to market because their systems are deployed more rapidly. The maintenance of critical systems is easier and more rapid as changes are made to the more easily analyzed model. The business can also choose to change its technology more readily as intelligence about the business is in the model not in the code. This means that organizations are no longer technology locked and can take advantage of a more flexible approach to delivery. The Legacy Renewal Solution from CA provides a new opportunity for businesses to rapidly take advantage of these benefits.

---

To learn more and see how CA software solutions enable organizations to modernize their legacy systems for better business results, visit [ca.com/legacyrenewal](http://ca.com/legacyrenewal).

CA, one of the world's largest information technology (IT) management software companies, unifies and simplifies the management of enterprise-wide IT for greater business results. Our vision, tools and expertise help customers manage risk, improve service, manage costs and align their IT investments with their business needs.

MP313580307