IBM Software
Rational

Extending the value of legacy applications through application transformation

Contents

Introduction

Business competitiveness and cost reduction are critical to the success of enterprises in today's business climate. Recent industry surveys reveal that the majority of CIOs are cutting discretionary spending, suspending capital projects and reducing headcounts.

IT costs are in the crosshairs of business executives, and CIOs are under enormous pressure to optimize their spending while continuing to deliver modern information services to help the business respond to competitive threats, grow market share and attract new customers.

Industry experts and IT executives alike recognize that a leading cause of high spending and limited business responsiveness are the very application systems that were developed over the years and that currently provide the information services that are the lifeblood of the business. These systems have high operation cost, do not take advantage of today's less expensive open computing environments, and are difficult to maintain, modify and extend. Therefore it comes as no surprise that the many industry surveys show legacy application modernization as one of the top CIO technology priorities.

Application modernization options include incremental re-architecting, redevelopment, outsourcing and divesting, purchasing commercial packages, and automated migration of existing applications to modern programming technologies that support a broad range of deployment options.

This paper explains why the choice of automated application migration is the least risky, the fastest for achieving tangible results, and typically the least expensive of the alternatives. The paper also describes the solutions that are available from IBM® Rational® software to achieve automated application migration.

The value of legacy applications

Business applications developed over many years have acquired a great deal of business knowledge it is fair to say that business data and applications are a digital image of the business processes. Even if they look aged and unappealing, their value in supporting the business operations is undeniable, as proven by the significant costs that companies have invested in their development, often in the hundreds of millions of dollars.

Although these applications can be a great asset, they can constitute a great liability—particularly with respect to innovation, business agility and, perhaps most important, cost of operation. When the liability reaches levels that executives consider damaging to the ability to grow their business, CIOs are challenged to seek alternatives.

The modernization alternatives

Early modernization approaches based on the creation of new UI layers atop the existing applications proved to be insufficient in addressing business agility and cost-of-operation problems. IT executives have been looking for more comprehensive alternative options, such as:

- Complete rewrite using modern technologies
- Acquisition and customization of commercial packaged solutions
- Migration of the existing code to modern technologies

The cost, risks, time to market and benefits of each of these options should be evaluated carefully before committing to an application modernization project.

A complete rewrite is often considered because it affords the highest flexibility in responding to business requirements, and it can be implemented using software development best practices, modular architecture, clean and highly maintainable code, and powerful development tools. However, a major undertaking such as a complete system rewrite is likely to be underestimated and fall behind schedule, over budget or deliver with functional shortcomings. According to the Standish Group CHAOS Summary 2009,[1] 32 percent of all projects succeeded, 44 percent were challenged (late, over budget, with less than the required functions) and 24 percent failed.

The typical challenges of a complete application rewrite include:

- High costs: With a simple estimate of the number of function points delivered by the existing system and the most optimistic development cost per function point, it is easy to see how a rewrite project can end up costing tens or even hundreds of millions of dollars.
- High risk: In a project of this magnitude the opportunity for underestimation, miscommunication and error is significant, adding to the overall risk of failure.
- Slow delivery: Application rewrite can take many years, and the problem of continuous rollup of incremental function being added to the system during the rewrite can further affect time to market.

The acquisition and customization of packaged applications is often considered a quicker and lower-risk solution, but some industry experts warn of the hidden costs and risk. According to Judith Hurwitz.[2]

"…packaged software is not really packaged. It is a set of tools, a set of templates and processes that are linked together based on marketing and promise."[2]

	Rewrite	Packaged Solution	Migrate
Cost	High	Medium	Medium/Low
Risk	High	Medium	Medium
Time to Market	Long	Medium	Low
Flexibility	High	Low	High

This chart contrasts the levels of cost, risk, time to market, and flexibility typically seen with a rewrite, packaged application solution and an automated migration.

Comparison of cost, risk, time to market and flexibility for three different modernization options

The reality of this choice is a potentially complex project with challenges similar to those of other software development undertakings plus:

- Hidden costs: In addition to the license cost and the development costs for customization and integration, which are fairly easy to forecast, companies can fail to recognize and budget for the time and effort devoted to the analysis of the package functionality and mapping to the business requirements, the testing effort, user training costs and business operation disruption.
- Limited flexibility in responding to business needs: Not having full control, ownership and know-how for the application source reduces the ability to quickly respond to new requirements from the business or react to competitive threats, not to mention the complete dependency on the vendor to correct critical errors that can affect day-to-day operations.

The third option is to migrate your valuable code into modern programming technologies using automated conversion tools. This can create the opportunity for your business to take advantage of new computing platforms and architectures while preserving the reliability and functionality of the existing system.

This option provides many of the benefits of the complete rewrite option with far fewer drawbacks. This approach gives IT organizations full control of the new system and enables the use of new architectures and tools, all in a shorter time and at a lower cost and risk than the other two options.

Using an automated conversion approach, the proven functionality that served the business so well for so many years remains essentially unchanged, helping to minimize disruption to business operations and to the IT development teams who make the transition to the converted system.

IBM Rational Migration Extension solutions

The IBM Rational Migration Extension (RME) family of products is a key component of the enterprise modernization strategy. With IBM RME products, customers can migrate applications that are built on dated languages with often expensive proprietary tools to a modern programming environment that enables deployment of applications and services to any modern open platform, eliminating the language-to-platform lock-in.

The family of RME solutions includes software and services for automated conversion of applications written in Software AG Natural and Adabas, CA COOL:Gen, CA Ideal for CA Datacom, RPG and COBOL. RME also includes the ability to automatically convert the applications' user interfaces, mostly text-based 3270 or 5250 screens, to state-of-the-art Web 2.0 interfaces that integrate seamlessly with the converted business logic.

The output of this conversion process is Enterprise Generation Language (EGL), an emerging business-oriented programming language from IBM that can be compiled into Java™, JavaScript™ and COBOL for deployment to modern runtimes such as Java Enterprise Edition (JEE) and Ajax-enabled Web browsers, or core runtimes such as CICS®, IMS™ and batch.

Automated conversion solutions from other vendors are designed to convert applications written in procedural languages directly to object-oriented languages. IBM's choice of EGL as the target makes it possible to achieve three critical goals that have proven to be elusive to the other approaches:

1. **Readability and maintainability:** The business nature of EGL enables a closer translation of original programming elements to similar ones, therefore the converted application will look familiar and easy to modify. EGL also eliminates the problem of code expansion: Applications written over many years are quite sizeable, often with several million lines of code, and any conversion that greatly expands the original size is not viable—the resulting application could reach tens of millions of lines of new code to maintain. EGL maps naturally, close to 1-to-1, producing converted applications of similar size of the original ones.

2. **Flexibility of deployment options:** EGL uniquely supports deployment to different tiers (browser, application server, data server) including high-performance batch processes, enabling virtually any application architecture while eliminating the need to maintain and extend the modernized system over a plethora of different technologies.

3. **Use of available developers:** Developers of all backgrounds have found EGL to be relatively quick and easy to understand and learn. Developers who have worked on the original application can become proficient in maintaining and extending the converted one, and new developers trained in modern programming technologies will become productive with a language notation that is powerful and familiar. This helps keep application maintenance costs in line with those of the premigrated system.

The converted application takes full advantage of modern middleware, such as Web and application servers and relational database systems. The converted application is also enabled for integration within a service-oriented architecture (SOA), and it can run on Microsoft® Windows®, UNIX® or Linux® servers, as well as all IBM platforms. The new application can be maintained using the Eclipse-based Rational development platform, boosting the productivity of the development team with state-of-the-art programming and collaboration tools.

RME migration process

Any application migration project based on Rational Migration Extension is implemented in three major phases:

1. **Discovery, analysis and planning:** RME migration projects begin with comprehensive discovery and analysis of existing source code and, where applicable, database files. The goal is to create a complete and detailed base of information about the system that is being converted. This phase identifies potential technical challenges that might be unique to your application, significantly reducing the risk of having unknown factors emerge during the actual code conversion process.

   The output of this analysis is a detailed application migration report that is used for project planning. This report provides insightful information about project deliverables and milestones, and enables accurate estimations of work effort, timelines and project costs.

2. **Code conversion:** Code conversion is performed on an off-site, secure server, typically requiring little contribution from the customer. Between 90 and 100 percent of the conversion can be automated, depending on the style and structure of the existing code. During this phase, the RME conversion tools read the source and convert it to EGL. These rules-based conversion tools can be customized and extended as needed. The resulting EGL code is both recognizable and maintainable, and uses the common libraries that are delivered in the Rational Migration Extension offerings.

3. **Implementation, testing and deployment:** The longest phase of any migration project takes place on-site, and the more resources you dedicates to this phase, the faster and more cost effectively it can be completed.

   This phase begins with the installation of the newly converted EGL source code into the EGL development workbench (for example, IBM Rational Business Developer), the RME software and, if needed, the new database schemas and loading of the client's data (such as when converting Adabas or Datacom/DB databases).

   Developers can then run the converted code through the EGL debugger or proceed directly to compilation and deployment by generating the final COBOL or Java (depending on the target platform) and HTML/JavaScript artifacts (if the migration project included transformation of user interfaces to EGL-rich Web interfaces).

   Finally, after all compiled objects have been deployed to a test system that simulates the production runtime environments, the converted system undergoes thorough testing (functional, system, performance and user testing). When all planned testing is concluded successfully, the application can be installed to the production system and the original application can be decommissioned.

Benefits and ROI of using RME for application migration

Applications that are migrated using RME can help IT organizations achieve new levels of flexibility and significant cost reductions.

The converted system can be extended in response to new business needs and requirements. Emerging technologies and standards can be incorporated easily to give users access to the information they need through intuitive and powerful interfaces.

The new code portability means freedom to standardize on the hardware and software of choice, reducing concerns about platform lock-in, duplicate costs, expensive processors, special high-cost proprietary software runtimes and additional administrative overhead.

Development teams can be more responsive and productive when using the modern programming language and tools and powerful collaboration infrastructure, which have been shown to reduce the development costs significantly.

IT executives can establish business process management in SOAs and rich user interfaces for better business agility and application investment return.

A modernization case study: Atos Origin

Atos Origin is a leading IT service provider specializing in consulting, system integration and outsourcing. Atos Origin clients wanted better service at reduced cost for their outsourced systems, as was the case for the merchandise information system that a leading German retailer outsourced to Atos Origin. This aging application, running in COBOL under IBM CICS Transaction Monitor on the IBM z/OS® platform, was considered too costly to operate and complicated for users. The retailer asked Atos Origin to implement a strategy that would cut operating expenses by nearly 30 percent and improve the usability of the system.

After evaluating possible alternatives, Atos Origin decided to convert the original system to an EGL code base that would allow rehosting the application to a Java runtime in the UNIX System Services (USS) environment on the scalable, reliable IBM System z® platform. This strategy would enable the use of less costly processing units that are available through the IBM System z Application Assist Processors (zAAP) specialty processor, with the aging user interface upgraded to a more modern, friendly look and feel.

Using IBM Rational Migration Extension software, the original code was converted quickly, replacing the original 3270 screens with graphical Web-based interfaces, and the business logic was regenerated into Java accessing the original IBM DB2® data on z/OS. The entire project was completed on schedule in eight months, and after two weeks of training Atos Origin developers were ready and the converted system went into production with no disruption. Users adjusted to the new system quickly, with virtually no training.

The application conversion strategy provided Atos Origin with an excellent competitive advantage, enabling them to offer their customer significant savings, a more usable system, and faster response to future needs. Atos Origin has quantified the savings up to 55 percent in operating costs with a return on investment of less than one year.

Conclusion

In the past, when IT organizations had to decide the fate of aging, mission-critical applications, their options were limited, high risk, and often disruptive, expensive and time consuming. An automated application migration changes these outcomes through lower costs and modernized, more powerful information systems to meet business demands.

About the author

Stefano Sergi

Stefano joined IBM as a software developer in 1980 and has held numerous positions within IBM Software Group in product development, product and market management, marketing and sales. He is currently the worldwide Application Transformation Solutions Manager for IBM Rational Software.

For more information

For more information about IBM Rational Migration Extension solutions, contact your IBM representative or business partner or visit:
ibm.com/software/rational/products/migration/

Additionally, financing solutions from IBM Global Financing can enable effective cash management, protection from technology obsolescence, improved total cost of ownership and return on investment. Also, our Global Asset Recovery Services help address environmental concerns with new, more energy-efficient solutions. For more information on IBM Global Financing, visit:
ibm.com/financing

IBM Corporation
Software Group
Route 100
Somers, NY 10589 U.S.A.

Produced in the United States of America
April 2010

[1] The Standish Group - http://www.standishgroup.com/newsroom/chaos_2009.php

[2] "Will packaged applications sink under their own weight?" http://jshurwitz.wordpress.com/2008/10/29/will-packaged-applications-sink-under-their-own-weight-five-recomendations-for-change/

Please Recycle

Rational. software

RAW14210-USEN-00