An Executive Guide to Oracle Modernization

Enabling Strategic Business Transformation

Learn More
Go to: oracle.com/goto/modernization
ORACLE IS THE INFORMATION COMPANY
An Executive Guide to Oracle Modernization

Enabling Strategic Business Transformation

Modernization allows organizations to maximize the use of their existing application assets as they move toward more agile and cost-effective technology environments.

IT organizations are striving to reduce total cost of ownership (TCO), improve their ability to react to changing business demands, and minimize their reliance on legacy skill sets—all while working to meet new compliance mandates. Modernization offers a holistic approach in which business goals, process, requirements, and TCO are central to the modernization of infrastructure and applications software.

**Legacy applications.** The average Fortune 100 company owns legacy applications comprised of 35 million lines of code. Generations of development teams evolved these applications to support a wide array of business-critical processes. This is one of IT’s biggest assets—and one of its greatest liabilities. Legacy applications monopolize resources: the average company spends 60 to 85 percent of its IT budget maintaining legacy applications and systems—systems that are too often inadequate to keep up with the changing needs of the business.

Given today’s flat or reduced IT budgets and a changing business environment, CIOs must make pivotal tradeoffs when allocating their budget—they must choose between maintaining existing assets or moving ahead with strategically important development and innovation. To make funds available for new business priorities, IT organizations must reduce the amount they spend on legacy applications and environments.

The solution? Oracle Modernization.

Modernization is an IT concept that is based on a fundamental requirement of doing business in the twenty-first century: maximize efficiency while increasing competitive advantage. Modernization is the process of evolving restrictive, legacy technologies to newer, open standards–based technologies—while retaining the business logic and data from the legacy systems.
**Why modernize now?** In almost every case, efforts to avoid or postpone modernization will make it harder to compete and, ultimately, to survive. The risks of not modernizing are significant. Aging legacy technologies are expensive to operate and maintain and adversely affect investments in new priorities—often the very systems that help generate new revenue or support current revenue streams more effectively.

The complex design of legacy systems is frequently based on outdated technologies, making them difficult to change and slower to respond to market shifts. In addition, maintaining legacy applications requires skills that are in short supply—programmers who know languages and systems such as Natural, COBOL, PL/I, Assembler, CA Gen, IMS, CICS, VSAM, and PowerBuilder are approaching retirement. Furthermore, legacy systems typically don’t track business processes well enough to satisfy the compliance mandates of today’s rigorous regulatory landscape—a limitation that could put company executives in jeopardy.

Forward-thinking organizations are already making progress in their modernization projects. For example, Office Depot is in the process of establishing business processes and creating software that it can use on a global basis. The firm has already decided that service-oriented architecture (SOA) is vital to its efforts because it will allow future applications to use a common set of reusable features or functions.

The U.S. Veterans Benefits Administration (VBA) has increased the flexibility and scalability of its compensation and pension processing system and has markedly improved its IT group’s responsiveness to new requirements. The VBA accelerated the rollout of customer service solutions for more than 3 million U.S. veterans and their families by migrating its legacy application from a mainframe to an open-systems platform that leverages Oracle Database, Oracle Tuxedo, and Oracle WebLogic Server. The back-end business logic is more reusable and scales more efficiently—from 11 million transactions per day to 16 million transactions per day—to meet new regulatory mandates of broader, multichannel access for more than 10,000 concurrent users.

German insurance provider WGV Group has reinvented its IT strategy by moving away from its outsourced applications and mainframe-based legacy infrastructure. It now uses Oracle JDeveloper and Oracle Application Development Framework with Oracle Database to support a new generation of SOA-enabled applications for core activities, such as policy generation and claims processing.
Approaches to modernization. Although modernization brings significant benefits, it can be a complex process. Inadequate assessment of application portfolios, insufficient planning, integration challenges, and employee resistance to new standards, software, and systems can all have a negative effect on modernization projects.

Successful modernization projects require substantial planning and investment to leverage best practices and proven methodologies. When properly structured, these projects can show a positive return on investment (ROI) in just a few years. Long term, they can continue to bring significant cost and agility benefits while enabling the business changes necessary to grow top-line revenue.

The modernization process begins with an assessment of an organization’s current environment and application portfolio and consideration of the company’s strategic and tactical requirements. After the assessment, a modernization road map is built that outlines the optimal mix of modernization approaches for each application to be modernized.

Modernization allows organizations to maximize their use of existing application assets as they move toward better technology environments. This is achieved through a mix of approaches.

- Replacing legacy applications with packaged, commercial off-the-shelf (COTS) applications
- Enabling service-oriented architecture (SOA)
- Rearchitecting legacy applications
- Automating the migration of legacy languages based on 4GLs and other legacy languages
- Offloading a mainframe’s MIPS transactions to an open system using data caching
- Rehosting application logic and data, intact, to a more-open, cost-effective, and agile platform

A combination of these approaches is typically required to provide an organization with a complete solution that addresses all their business needs.

With computer systems an integral part of today’s business success, deciding not to move to modern technologies is no longer an option. Organizations that do not begin the move to modernization will find their business lagging behind those that are aligning their IT strategy with business goals in a cost-effective manner. Modernization reduces costs, increases agility, improves compliance, and reduces reliance on disappearing legacy skill sets.
Learn More
Go to: oracle.com/goto/modernization
Table of Contents

MAGAZINE ARTICLES
10  Growing Green
16  Moving Forward
23  Modern Thinking

SUCCESS STORY
29  Up to Date, Up to Speed

WHITE PAPERS
33  Why Modernize?
33  Introduction
34  Why Modernize?
36  » Reduced Total Cost of Ownership
36  » Increased Agility
36  » Reduced Reliance on Legacy Skill Sets
36  » Improved Compliance
37  Why Oracle Modernization?
37  » Oracle Delivers for Customers with Legacy Software
39  Conclusion

41  Approaches to IT Modernization
41  Introduction
42  Approaches to IT Modernization
42  » Replacing Legacy Applications with Packaged Applications
43  » Enabling Service-Oriented Architecture
44  » Rearchitecting Legacy Applications
46  » Automating Migration of Legacy Applications
50  » MIPS Offload
50  » Rehosting Legacy Applications to an Open Platform
54  Conclusion
Magazine Articles

Learn More
Go to: oracle.com/goto/modernization
GROWI

BY MARTA BRIGHT
What does “going green” mean to organizations that must handle more information than ever before? Developing the next generation of clean energy production? Hoisting solar panels onto the roof? Rethinking the design of the data center? Server virtualization?

“Embracing green practices involves a little bit of all these things,” says Nigel Montgomery, research director at AMR Research. Two organizations—one involved in energy research, the other in energy supply—are using Oracle technology in different ways to achieve the same goal: improving the world’s environmental health.

KEEPING A SAFE YET EFFICIENT DISTANCE

Pacific Gas and Electric Company (PG&E) provides power to 15 million people scattered throughout 70,000 urban and rural square miles of Northern and Central California. Delivering energy around the clock to its customers is a balancing act that has the utility purchasing, generating, distributing, renewing, and reducing the outflow of energy when customer demand peaks and power grids become overtaxed. To do this better, PG&E is spending nearly US$1 billion in enhanced demand response and energy efficiency programs. The utility hopes that this effort, the largest IT project in the company’s history, will challenge customer thinking and behavior.

One of PG&E’s premier efforts, the SmartMeter program, is a usage-monitoring plan that eventually will reach across the company’s entire service territory and give customers detailed rate and usage information that will help them understand, manage, and reduce their gas and electric consumption.

“Over the next five years, we’ll have computer
“Much of what we’ve accomplished we attribute to Oracle RAC. Instead of scaling up a massive mainframe, we’ve distributed across smaller, more energy-efficient systems.”

—Eugene Park, Senior Director of Application Services, PG&E

modules on literally every meter,” says Eugene Park, PG&E’s senior director of application services. “This will alter a once-a-month customer relationship to a once-an-hour—potentially once-every-15-minute—relationship.”

The benefit is instant information that’s gathered, analyzed, and then sent back to customers (possibly by e-mail or cell phone alerts) as a reminder to, for instance, run their washing machines and vacuum cleaners after 7 p.m. when demand and prices are lower. “SmartMeter technology helps us show customers exactly when and how they can save money and energy by backing off usage during peak hours,” says Park.

As PG&E expands the SmartMeter program, the utility will also be phasing out onsite meter readers, which means fewer carbon-producing company vehicles out on the roads. “In addition to regular readings, if customers want to stop or start service, SmartMeter technology will enable us to operate remotely instead of having somebody drive out there to reconnect service,” says Park.

The SmartMeter program started gaining traction in 2005 when PG&E wanted to enhance its customer care and billing system, which produces 350,000 bills and processes US$60 million in payments each day. “We needed a substantial upgrade,” says PG&E IT Director Alain Erdozaincy. At the time, the utility was using an IBM DB2 database operating on a mainframe, which PG&E needed to leave behind.

“We knew we needed to support the data growth created by the SmartMeter program, so we worked with Oracle on creating a scalable clustered configuration,” says Erdozaincy. PG&E now depends on the Oracle technology stack—including Oracle Database, Oracle Real Application Clusters (Oracle RAC), and Oracle Grid Control—to run its data center.

With Oracle RAC, the utility can use smaller, more-efficient servers that offer excellent performance and scalability while handling 720 times the amount of data, according to Park. “Much of what we’ve accomplished we attribute specifically to Oracle RAC. Instead of scaling up a massive mainframe,
we’ve distributed across smaller, more energy-efficient systems,” he says.

That approach makes sense to AMR Research’s Montgomery. “If you’re using processing time, you’re using energy,” he says. “Energy costs money, so if you can increase efficiencies and reduce processing time, the savings can be put to better use—focusing on other IT- or non-IT-related activities such as improving energy consumption in other parts of the business.”

PG&E also offers a variety of programs that will reward corporate customers for adopting server virtualization/consolidation along with other improvements such as enhanced airflow control systems and more-efficient data storage.

STILL WATERS, DEEP ENERGY
Generating power from nuclear fusion could become one way to dramatically reduce the carbon footprint. “Thermonuclear fusion is the gold standard of clean-energy-related technologies,” says Edward Moses, principal associate director, National Ignition Facility & Photon Science (NIF) at the Lawrence Livermore National Laboratory. “People might not be completely knowledgeable about how fusion works at this time, but it may be the solution to the growing clean-energy production challenges that we are all deeply concerned about,” he says.

Scientists and engineers at NIF are using extremely powerful lasers to produce controlled thermonuclear fusion. By squeezing isotopes of hydrogen atoms to intensely high pressures and temperatures similar to those that exist at the center of the sun, they hope to create more energy than what was put into the system. The laser system—which at full capacity will fire 192 beams—uses up to 60 instruments such as cameras and oscilloscopes to measure the results of running an experiment. (This work was performed under the auspices of the Lawrence Livermore National Security, LLC [LLNS], under Contract No. DE-AC52-07NA27344.)

“Even with all of this interplay of instrumentation, an experiment is over in less than a millisecond of a second; in fact, most of the data is taken in a few billionths of a second. What we’re left with are almost unimaginable amounts of data that have to land in our Oracle Database and be analyzed within 30 minutes,” says Tim Frazier, NIF applications director, who is responsible for managing scientific data.

To manage its business as well as its science, NIF runs Oracle Database 11g with Oracle RAC, Oracle Application Server, and the Oracle content management framework. “We use Oracle in all areas of the business, from construction planning and budgeting down to the database-driven laser control system,” says Frazier.

With just a 30-minute window of time in which to capture and store massive amounts of image-based data, the performance of Oracle Database 11g—and its Oracle SecureFiles feature—is critical. “If we don’t have a high-performance method for saving images into the database, the instruments may time out, and we’ll lose data,” Frazier explains. “When we’re in full operation, we’ll generate hundreds of terabytes each year, which means that the compression feature of [Oracle] SecureFiles will also play a critical role in helping us economically manage all of the associated storage.”

As for year-over-year data growth, according to Frazier, the intention is to keep all experimental data and the results of analysis available for instant retrieval throughout the lifetime of the facility, which is approximately 30 years. “Add to that all...
of the metadata that accompanies our experimental data, and in just a three- or four-year time frame, we could easily reach the multipetabyte level,” he says.

The analytic phase of fusion experiments is driven by a series of Oracle BPEL workflows that perform tasks such as shot scheduling, supervising and coordinating the schedule and flow of shot data analysis, maintaining the requisite models for analytical functions, and providing data services for the modules. Oracle BPEL Process Analytics provides a user interface that allows scientists and engineers monitor the analysis process and address anomalies if and when they arise.

“Through the miracle of Einstein’s $E = mc^2$, the excess energy produced, which has no carbon waste byproducts associated with it, can be used to generate consumable power,” says NIF’s Moses. NIF will be the first facility in the world that will have the laser capacity necessary to do this work, although the overall objective of NIF, as a research facility, isn’t to generate power. “We want to help engineers and scientists understand how to generate power with fusion energy,” Moses says.

The potential benefits are limitless. “Through NIF we can potentially bring about a supply of clean energy that, if tapped, could be used for humanity’s benefits far into the future. That’s a very exciting possibility,” says Moses. “Companies like Oracle are key to succeeding in what many consider humankind’s grand challenge mission of clean and plentiful energy.”

Marta Bright is a senior editor with Oracle Publishing.

**next STEPS**

READ about
- Oracle Advanced Compression
- Oracle VM
  [oracle.com/technologies/virtualization](http://oracle.com/technologies/virtualization)
- Oracle Utilities Customer Care and Billing
- Oracle Real Application Clusters
  [oracle.com/database/rac_home.html](http://oracle.com/database/rac_home.html)

DOWNLOAD Oracle Database
[otn.oracle.com/software/products/database](http://otn.oracle.com/software/products/database)
The world of information technology moves relentlessly forward, and many CIOs find themselves in a quandary. Some of their legacy information systems were created when programmers took it upon themselves to automate processes in the workplace using COBOL, IP Multimedia Subsystem, virtual storage access method (VSAM), Natural, and a wide variety of otherNow-aging technologies and tools.

As much as 80 percent of IT budgets are spent operating and maintaining these legacy applications, and this percentage is projected to grow in the next few years. That doesn’t leave much budget for new development, let alone the type
FORWARD

LEGACY SYSTEMS AND SPEEDS THE ENTERPRISE TO SOA.
of IT transformation required to take advantage of open systems, service-oriented architecture (SOA), and process-driven workflow systems.

While companies are challenged to embrace open standards, internet computing models, and SOA, they also can’t afford to throw away the mission-critical content of their entrenched business systems. And standing still is not an option for businesses as the competitive pressures to become more agile and move to cost-effective IT architectures continue to grow.

MOVING FORWARD

Some organizations are moving their business operations away from aging applications and into more-flexible and more-versatile IT environments—a process called IT modernization. These organizations are modernizing by retaining existing application asset content while transforming these assets to modern languages, database systems, and SOA services. The tricky part, however, involves preserving the business content of these applications as they are transformed to the new environment. This is especially important when the business knowledge encapsulated within legacy applications is unknown outside of those applications.

“We’ve seen a lot of organizations looking at how they can modernize their applications,” says Rebecca Wettemann, vice president of Nucleus Research, a firm based in Wellesley, Massachusetts.

“As companies look to the next generation of enterprise applications, they are trying to design solutions that streamline management, require less administration, and require less troubleshooting, while providing more availability to end users,” Wettemann says.

Most modernization efforts begin with an assessment of the legacy application portfolio to determine the state of current systems, which applications are the best candidates for modernization, and which modernization techniques will deliver the greatest returns. A big part of this process is a frank assessment of the IT workforce. People with skill sets in legacy technologies such as COBOL, Adabas, and integrated data management system (IDMS) are simply getting harder and harder to find.

The next step is to examine the best modernization approach. Legacy modernization can be approached several ways, including SOA integration, rehosting, automated migration, commercial off-the-shelf (COTS) replacement, and rearchitecting.

ON TRACK FOR AUTOMATED CONVERSION

The Danish Commerce and Companies Agency (DCCA) decided to modernize an IBM MVS mainframe system that processes more than 350,000 business registrations each year. As part of the Ministry of Economic and Business Affairs in Copenhagen, the DCCA is responsible for regulating companies, accountants, real estate agents, authorized translators, interpreters, restaurants, and hotels.

According to David Graff Nielsen, a project manager and IT architect at the DCCA, their environment was expensive to maintain and didn’t easily integrate with the agency’s newer information systems. “We absorbed high facility management costs, partly because our registrations were scattered over a large number of dissimilar systems and platforms,” he says.

The DCCA hired Oracle modernization partner BluePhoenix Solutions to transform this IBM MVS system from an OS/390 environment to an Oracle/Java Platform, Enterprise Edition (Java EE) environment on a Linux platform. The agency’s goal was to maintain the level of functionality provided by the existing system while adopting a more modern and flexible software platform—and to do the migration automatically.

“The advantages of automated migration are speed and consistency,” says Lance Knowlton, vice president of modernization solutions at Oracle. “Since a computer carries out the conversion process, it can be done quickly and consistently.
"The automated migration is a step toward a more radical rearchitecting of all of the agency’s registration systems. The agency plans to rework the modernized application as it gradually adopts SOA."

—David Graff Nielsen, Project Manager, IT Architect, Danish Commerce and Companies Agency

and can even be repeated on a more recent copy of the source code to include ongoing changes.” However, Knowlton says that as with all automated tasks, “these types of conversions are most successful when there is a well-defined and understood mapping between the source and target architectures.”

Another prerequisite for a successful automatic migration is high-quality source code. In the DCCA’s case, BluePhoenix took an inventory to determine the scope of the migration effort and identify obsolete code. Then the modernization team created a rule set for converting Adabas to Oracle Database and Natural to Java. Next, they used BluePhoenix’ DBMSMigrator for Adabas/Natural to migrate the Adabas/Natural applications to a relational system, with careful attention to those specific constructs such as reinput statements as well as all Natural objects including programs, subprograms, maps, data area types, and so forth.

In all, the DCCA migrated more than 2,700 Adabas/Natural programs, totaling 1 million lines of code, and 200 Adabas file
“We have a long-term strategy for moving to a more open environment, and new projects are exploiting opportunities to redevelop existing code. . . . Halving IT operating costs on the modernized application infrastructure has marked the project as a huge success.”

—Brian Henderson, Head of Technical Design, Royal London Group

views. Additionally, the DCCA converted 1,000 Natural character-based screens to graphical Web screens. The new system has been working well in production for more than a year. “About 80 percent of the code was converted without issues, and the new applications work as expected,” says Nielsen.

In a typical month, the DCCA runs 800,000 transactions over its new applications. Since adopting the new software platform, managers have seen a 50 percent reduction in application operating costs, and they expect to achieve a complete return on investment in three to five years, Nielsen reports. The DCCA is churning these savings back into the business by accelerating the deployment of Web-based services, leading to a faster, safer registration process.

“The automated migration is a step toward a more radical rearchitecting of all of the agency’s registration systems,” Nielsen says. “The agency plans to rework the modernized application as it gradually adopts SOA.”

OLD LOCOMOTIVE, NEW TRAIN
Another approach to modernization, called rehosting, involves migrating legacy application code to a modern underlying database and hardware platform while leaving the application logic

Brian Henderson, head of technical design at Royal London Group, says that the company looked to modernize its IT infrastructure when its existing systems had a hard time handling an increasing number of transactions while delivering a high quality of service to internal and external customers.
largely untouched. This is done by using a layer of software that looks like the legacy environment to the application code but in actuality is running on an open systems platform.

This was the strategy selected by Royal London Group, which as the U.K.’s largest mutual life insurer manages more than US$55 billion in funds for more than 3.5 million customers. Royal London depended on Unisys mainframe systems to run its life assurance and pension administration systems. These business systems handled hundreds of thousands of transactions per day, supported by overnight batch runs.

According to Brian Henderson, head of technical design at Royal London Group, the mainframe systems were having a hard time handling an increasing number of business transactions while delivering the quality of service internal and external customers required. Henderson’s team also wondered how long the underlying technologies would be supported and enhanced by Unisys. “Due to the size and complexity of the applications, we were encountering technical and capacity limits, resulting in batch process overruns and unacceptable downtime for online users,” Henderson says.

The unpalatable cost of upgrading the Unisys hardware and software environment, coupled with the underlying infrastructure issues, led Royal London Group to look at modernization options, including rearchitecting, rehosting, and COTS replacement.

According to Nucleus Research’s Wettemann, many companies are adopting a “plain-vanilla, out-of-the-box” strategy as much as they can. “We see a lot more organizations looking toward the vanilla approach just because it’s so much less risky and less costly to develop and deploy. It’s also less costly to upgrade in a lot of cases,” she says. “But what they can do with integration technology and solutions like Oracle Application Integration Architecture, for example, is take those pieces that are very specific to their industry or even their business model and link them into that broader architecture that is more horizontal in nature.”

Royal London ultimately decided to rehost the code because the firm believed it would be the least risky and most cost-effective alternative within the required timeline. The company engaged with HP as the hardware and infrastructure provider; Oracle as the database vendor; and MSS, an HP and Oracle modernization partner that provided the automated toolset to translate Unisys EAE and COBOL code to C, PL/SQL, and Micro Focus COBOL on a UNIX platform.

The project took 18 months to complete, culminating in a switch from the Unisys system to the Oracle/UNIX system over just one weekend. At that time, the team rehosted 10 million lines of application code and transferred 750 million data records to the Oracle database.

“The vast majority of the migration was automated using the MSS toolset,” Henderson reports. “However, there were a small number of certain scenarios, particularly in the COBOL arena, where we had to write scripts and incorporate them into the migration process.”

The new business systems are hosted by 12 HP servers, and Henderson is pleased to report significantly improved performance from the Oracle/UNIX environment. “We have reduced the overnight batch runs from 11 or 12 hours to 4 or 5 hours, which allows us to process business over an extended working day,” he says.

While most of the rehosting process went smoothly, Henderson admits that some of the rehosted mainframe code remained difficult to change and test in the new environment. “We have gradually reduced our reliance on the legacy migrated code, but it has taken longer than expected,” he says.

“However, we have a long-term strategy for moving to a more open environment, and new projects are exploiting opportunities to redevelop existing code,” he adds. “On top of this, achieving a return on investment within two years and halving IT operating costs on the modernized application infrastructure has marked the project as a huge success.”

**STOKING THE ARCHITECTURE**

Many organizations today feel so compelled by the importance of getting the full value of modern IT architectures and open standards that they choose to rearchitect their legacy systems. Rearchitecting a legacy environment allows businesses to retain business-relevant processes and assets from legacy applications while eliminating all dependence on legacy technologies. Rearchitecting is common for modernization efforts that require fundamental changes in the design of the application to adopt an SOA architecture including the full use of new technology areas such as business intelligence and business rules engines.

The Government of New Brunswick, Canada, chose this approach to modernize its Medicare system, which processes between 6 and 7 million physician claims per year. The current Medicare system, which was implemented in 1982 by the New Brunswick Department of Health, is based on a Unisys mainframe system running an OS 2200 operating system and a DMS 2200 database.

The Department of Health decided to vacate the legacy environment and move to more-modern and cost-effective tech-
nologies. The agency considered a number of alternatives to rearchitecture, including outsourcing the Medicare system to a private consortium.

The Department of Health began by analyzing the legacy environment through a set of tools and processes provided by Oracle modernization partner Make Technologies. During phase one, the application's operational and functional characteristics were documented to establish a baseline for the new Medicare application. Then the agency used Make's Transformational Legacy Modernization technology, which involves extracting the intellectual property residing in the existing application and transforming the legacy data, assets, and processes into a Web-based, Java EE SOA application based on an Oracle database.

Travis Foster, project manager at the Department of Health, says that the Medicare project is progressing as expected, despite an initial limited understanding of the embedded logic in Medicare's legacy source code. “Moving away from the legacy character-based screens to more-streamlined graphical user interfaces will allow Medicare staff to perform their daily tasks more efficiently,” he notes. “It will be easier to train new employees and less expensive to maintain the system.”

Having extracted the core content of the existing environment, Foster and his colleagues are determining the final design for the new rearchitected Medicare system, a process that will be completed in May of 2008. The Transformational Legacy Modernization methodology and toolset will then be used to rearchitect the core content into the new design. The projected payback for the Medicare modernization project is five years, based on annual savings in maintenance costs of CA$1.6 million.

STAYING ON TRACK

Undertaking these modernization projects has both pros and cons. One of the biggest drawbacks comes with the extra responsibility of taking on a mission-critical long-term project and commitment. That's why Foster advises careful preparation: collecting existing documentation and setting a clear strategy and vision before moving forward. "There is no silver bullet when doing a legacy modernization project," he says. "Success is dependent on the organization committing the necessary resources."

Royal London Group's Henderson agrees, saying that the key lesson his team learned was to invest as much effort and time as possible in an initial proof of concept with the technology partners. "This will pay big dividends in the overall program timeline," he says.

In the end, each organization must determine its own modernization road map based on its own specific business drivers, legacy technologies, aversion to risk, and other business factors. But once the determination has been made that modernization makes sense, the important thing is to get started.

"While some organizations might make the mistake of thinking that there is no compelling need for modernization, doing nothing can be a recipe for disaster," cautions Oracle's Knowlton.

"Today’s newer generation of IT staff is not skilled in legacy environments, and this situation will likely get worse," he says. "Better to invest today, while there is still time to define an optimal modernization road map, than to find yourself with a bigger problem tomorrow."

David Baum (david@dbaumcomm.com) is a freelance business writer based in Santa Barbara, California.

"Moving away from the legacy character-based screens to more-streamlined graphical user interfaces will allow Medicare staff to perform their daily tasks more efficiently," says Travis Foster, project manager for the New Brunswick, Canada, Department of Health.
Modern Thinking

MODERNIZING LEGACY SYSTEMS IS A MAJOR IT COMMITMENT, BUT SIGNIFICANT COST SAVINGS, GREATER AGILITY, AND LOWER LONG-TERM RISK ARE THE REWARDS.

When Office Depot replaced its merchandising system in 2005, swapping an aging homegrown application for the Oracle Retail Merchandising System (formerly Retek), it was just the tip of the iceberg. The migration from a legacy application to the more-open architecture of Oracle’s software helped Office Depot significantly increase competitiveness, grow top-line revenue, and slash IT costs; but just as important, it opened the company’s eyes to the need for a more-expansive IT modernization project. Now the US$15 billion-a-year retailer is taking stock of its entire application portfolio and its underlying technology infrastructure and is intent on modernizing aging systems and doing away with redundant ones.

Modernization is a fast-growing IT concept that’s based on a fundamental reality of doing business in the twenty-first century—namely, to remain efficient and competitive, companies need to replace restrictive legacy technologies with newer, open standards–based technologies while retaining the business content stored in those legacy systems. Many large companies still depend on mainframes or, in the case of Office Depot’s old merchandising applications, outdated application development platforms such as Sybase PowerBuilder that have been in place 15 years or longer. What’s more, few programmers possess the skills needed to handle the code rewrites to adapt those technologies to changes in the market.
However, once they switch out their hard-coded mainframe environments with standards-based application infrastructures, companies typically realize significant cost savings, reduced dependence on hard-to-find programming skills, and newfound flexibility and agility. Such is the potential of IT modernization—something that became abundantly clear to Office Depot during the company’s merchandising system upgrade.

“That upgrade has led to a much broader modernization effort that we have underway today,” says Mike Kirschner, vice president of IT for the Delray Beach, Florida–based office supplies giant. “We have a lot of local systems or regionalized systems now, so one of the things we’re trying to do is create a set of software that we can use on a global basis. That’s going to require retiring some legacy components in some cases. In other cases, it’s not just legacy systems. We have to make choices to retire even some of the more-modern software just because we don’t need two copies of different software doing the same thing.”

**FUTURE PLANNING**

Kirschner and his team are turning over every rock—from enterprise resource planning and finance to shipping and fulfillment—to prepare Office Depot for the fast-changing business environment needed to remain competitive for the next decade. “Some applications, such as human resources, where we recently upgraded from a homegrown application to [Oracle’s] PeopleSoft, are fine. For other applications, we clearly need an exit strategy and a new application,” he says. And just as Office Depot needed to move away from relying on PowerBuilder, the company is taking a hard look at what kind of infrastructure its next-generation applications will run on. The company’s new merchandising system runs on the Sun Solaris Operating System, making use of a newly deployed Oracle database. Although Office Depot hasn’t committed to any new technology purchases, that foundation, along with the company’s focus on working with vendors that support open standards, gives Oracle a clear advantage.

“The first thing we’re trying to do is establish what our global business processes are. How do we want to run our business—not just how do we run it today in North America or Europe, but how do we want to run it tomorrow on a global basis?” says Kirschner. “As we prioritize our needs, we’ll figure out what application areas to focus on. We’ll also do an analysis around where our pain points are with our legacy applications.” Only then will Kirschner’s team make technology choices and begin modernizing additional applications.

One thing Kirschner is certain of: a service-oriented architecture (SOA) will be a vital piece of the puzzle, allowing the company’s future applications to make use of a common set of features or functions. “If you just face the fact that you need to have integrations, you want to figure out how to best support those integrations in a reliable, secure, and scalable fashion,” he says. “We think service-oriented architecture is a good pattern to use for those integrations.”
“One of the things we’re trying to do is create a set of software that we can use on a global basis. That’s going to require retiring some legacy components.”

—Mike Kirschner, Vice President of IT, Office Depot

MODERNIZATION CHALLENGES
If the whole modernization process sounds complex, that’s because it can be. Inadequate assessments of application portfolios, lack of employee adoption of new systems, unsuccessful integrations, or even a simple drying up of funding can all negatively impact modernization projects. And even successful modernization projects require substantial investment.

But in almost every case, efforts to avoid or postpone the effort of modernization will come back to bite IT management. The risks of not modernizing are significant. Aging legacy technologies are expensive to operate and maintain, draining IT departments of precious dollars that could be spent on systems that help generate additional revenue. Their complex design based on legacy concepts makes it difficult to implement changes, thus increasing the challenge of responding to shifts in the market, and they require skills that are in short supply as the market, and they require skills that are in short supply as the environment and are in a state of denial.

Many customers look at their legacy environment and are in a state of denial. The systems haven’t fallen down, and they know that it can be expensive for them to take the systems and move them forward. But we know that at some point, these systems are going to cause problems.”

GETTING OPEN
Take, for example, the insurance industry, where the mounting challenges posed by legacy environments made the need to modernize apparent several years ago. Many insurance companies began looking for packaged applications to replace core legacy systems that handled claims processing, but their searches failed to identify anything that met their needs, so they stopped looking.

“Five, seven, eight years later, their applications have declined, and they’re back to thinking, ‘Now I need to go ahead and do something about them, but now I’m eight years closer to the retirement of all of the skill sets that I’ve got,’” says Dale Vecchio, research vice president, Gartner.

That is precisely the scenario that Oracle—with the help of a bevy of partners such as Accenture, CSC, EDS, HP, Perot Systems, and Unisys—is trying to prevent. While Microsoft tries to convince large companies to move to its .NET architecture and IBM attempts to keep its customers on its proprietary mainframe platform, Oracle has opted to help customers formulate a strategy for getting from point A to point B. In other words, as long as customers take the initial step of migrating to open architectures, there will be opportunities to promote a whole universe of standards-based applications, middleware, and supporting infrastructures.

That approach has been packaged in a no-cost service dubbed Oracle Modernization Insight, in which a team of Oracle modernization experts works with a customer to formulate a realistic strategy for establishing a standards-based, process-driven SOA environment and determine a road map to get there. Naturally, in its role as modernization consultant, Oracle educates customers on its applications and technology stack for implementing modernized applications—from its suite of service-enabled applications to Oracle Fusion Middleware, Oracle Database, and third-party hot-pluggable SOA-based products—and makes sure they understand that the Oracle products can support even the most demanding business processes.

“A lot of customers believe that the horsepower that exists on their mainframe cannot be reproduced in an Oracle environment, and that is just simply not true,” says Knowlton.

Take German insurance provider WGV Group. When the European insurance industry was deregulated in 1993, the company quickly deduced that in the newly competitive market, it could no longer depend on its antiquated systems
“The forces that are impacting modernization are often out of your control, but you can’t ignore them.”

—Dale Vecchio, Research Vice President, Gartner

ted by global positioning system—enabled monitors installed in cars, allowing WGV to set rates based on actual driving behavior rather than demographics. The company is also planning to roll out a new analytics application that will be used to improve vendor relations and is streamlining the processing capabilities of its e-invoicing system for auto repairs.

Make no mistake, though; not every IT department acts with WGV’s proactive vision. Instead, they allow shrinking IT budgets to prevent them from committing to big-ticket projects. They see a fast-changing business environment and are paralyzed rather than energized. They look at their diminishing legacy programming expertise and are unsure how to minimize that skills gap. And they watch helplessly as their antiquated systems prevent them from complying with an ever-expanding sea of regulatory requirements. Instead of tackling those issues through IT modernization, they employ a patchwork approach to tweak their existing systems. But they do so at their own peril. “The forces that are impacting modernization are often out of your control, but you can’t ignore them,” says Gartner’s Vecchio.

And the opposite problem happens as well—too often companies realize the risk but plunge into modernization projects hastily. They embark on poorly planned efforts that are doomed from the start, when, in fact, the road to IT modernization is paved with preparation and strategic thinking. “In the worst cases,” says Vecchio, “you spend the money and you never get what you were looking for; in the best cases, you have significantly reduced the cost of IT, and your ability to respond is much, much greater.”

There’s not a CIO around today who won’t see the value in that, but how they make the trip will determine whether they, and their businesses, will succeed.

TOBY KONTZER is a freelance business technology writer based in Silicon Valley. He recently concluded a six-year stint at InformationWeek, and over the years he has written for Investor’s Business Daily, CIO Insight, Network World, and Wired.
Up to date, up to speed
How a three-way industry partnership helps to cut the specter of modernization down to size.

When San Francisco’s famous Bay Area Rapid Transit (BART) system decided to move away from mainframe computing, its decision reflected a growing trend. BART’s objectives: new applications, automated processes, and a more flexible infrastructure with integrated, accessible applications and information, enhanced disaster recovery and business continuity safeguards, and an easier path for future upgrades.

The idea is catching on. In a 2008 study, Penn, Schoen & Berland Associates found that 57 percent of those surveyed were planning to move all or some applications off mainframes. * But the thought can be intimidating. How can you modernize without disrupting the business? How do you choose new infrastructure? Where can you find resources, tools and people with modernization experience to help you take the first steps?

In response to the growing need, HP has helped to found the Application Modernization Initiative (AMI), a three-way industry partnership between HP, Oracle® and Intel® that helps customers identify applications that can be modernized from a legacy mainframe environment onto more modern open infrastructure. HP participates in AMI through the amalgamation of two related programs: the Mainframe Alternative Program, which helps enterprise customers better understand how to transition from mainframe systems, and the Application Modernization Business, which supports the modernization of legacy applications.

“There are three reasons people move away from mainframes,” says John Pickett, Manager of the HP Mainframe Alternative Program. “The first is agility. Legacy applications on a mainframe just aren’t as agile as those running on an open infrastructure. Being unable to quickly adjust your mainframe applications to respond to competitive pressures disrupts the whole business.”

The second reason, Pickett says, is cost. Mainframes are expensive, and even the new processors to help them cope with today’s workloads can cost hundreds
of thousands of dollars. Finally, he says, mainframe skills are on the wane. IT managers are facing a shortage of the skills required to support and manage legacy applications and systems. Modernized infrastructures don’t face these challenges.

The modernization map
HP’s contribution to AMI includes Application Modernization Services together with HP Integrity servers, working within the Virtual Server Environment (VSE) Reference Architecture. The HP Integrity servers run on Intel® Itanium® processors. Oracle’s Grid Computing Platform and SOA capabilities delivered through Oracle Fusion Middleware (OFM) are also a vital element, and include Oracle® Database with Real Application Clusters and Oracle® Enterprise Manager/Grid Control.

“In making all of these elements work together, HP, Oracle and Intel offer a complete, integrated modernization roadmap,” says Sumanth Tarigopula, Director of HP Modernization Services. “We call it the ‘integrated stack.’ Risk is removed because our testing and validation have been done ahead of time, rather than at the customer site.”

In BART’s case, Oracle’s PeopleSoft Enterprise software provided a quick return on modernization, meeting most of BART’s 2600 software requirements ‘out-of-the-box,’ with minimal customization. BART chose to support the software with an infrastructure based on HP Integrity servers.

“We like the architecture and future roadmap of HP Integrity servers, Intel Itanium 2 processors, and the HP-UX 11i operating system,” says Robin Cody, BART’s Department Manager of Information Technology. “They will provide us with more capacity, scalability and unlimited growth potential. They have allowed us to bring in applications that couldn’t run on a mainframe and rebuild BART’s business.”

A step-by-step approach
Customers considering modernization know it has to be done, but the scale of the work can still be intimidating. That’s why the AMI team sits down with the customer to work out which part of the business will benefit the most.

AMI offers five approaches to modernization, based on the unique requirements not only of each customer, but of each system. Modernization may mean replacing a legacy application with a standardized one. Or perhaps it’s time to simply retire the application, because it just isn’t being used or is redundant.

The third option is re-hosting: here hardware is identified as the performance barrier, and the application is moved to a more efficient infrastructure.

The fourth approach is to re-engineer a failing application to perform as required. Lastly, legacy apps can be retained as is while other priorities are addressed. This strategy helps to break modernization down into straightforward, understandable procedures, and the synergy contributed by the AMI partners working together makes the process faster, simpler and more effective.

“Customers who know they need to modernize often hold back because they’re not sure how to move forward,” Pickett says. “That’s why it’s such a good idea to combine HP’s infrastructure and services with Oracle’s enterprise capabilities and the processing power of Intel. It gives the customer a broad range of infrastructure, tools and services, and that breadth frees the team to focus on what the customer needs.”


This article first appeared in the Spring 2008 Main Edition of HP’s “Transforming Your Enterprise” Magazine. The current edition of this magazine is available at: www.hp.com/go/transform

To download a white paper The Application Modernization Initiative, visit: www.hp.com/go/transform

© Copyright 2008 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

4AA2-1688ENA, August 2008
All organizations want to realize the highest business value possible from their existing investments in IT infrastructure and application software. However, maintaining the application and infrastructure software in a legacy environment consumes a disproportionate percentage of IT budget and human resources.

The average company spends from 60 to 85 percent of its IT budget maintaining legacy applications that fail to meet the changing competitive needs of the business. As a result, IT organizations are under increasing pressure to reduce costs and react more nimbly to ongoing business demands. The solution? IT modernization.

IT modernization is the continuous evolution of an organization’s existing application and infrastructure software, with the goal of aligning IT with the organization’s ever-shifting business strategies. Such alignment enables top line revenue growth, while reducing bottom line expenses—resulting in increased profit margins. IT modernization also lets organizations maximize their existing application assets as they move toward a more open, complete, and integrated application and infrastructure platform.

Organizations that do not begin the move toward IT modernization will soon find their business lagging behind their competitors.
IT modernization is the continuous evolution of an organization’s existing application and infrastructure software, with the goal of aligning IT with shifting business strategies. IT modernization implies the acquisition and deployment of modern technologies—along with their associated skill sets and capabilities—to replace legacy environments.

Why Modernize?

Legacy applications have become a significant business problem. They carry a high cost of ownership, are difficult to modify to meet ongoing business demands, require a legacy skill set that fewer and fewer people possess, and do not adequately meet today’s compliance demands.

For these reasons, organizations are considering the move to new technologies and architectures. But while it is possible to develop applications from scratch that fully utilize new technologies, the approach is expensive and risky. The strategy that a growing number of organizations are embracing is to modernize their existing applications and infrastructure software.

To get the maximum strategic business benefit from modernization, it is important to base the modernized system on an architecture that is built on open standards and deployed on open systems. IT modernization based on an open architecture offers the benefits of reduced total cost of ownership, increased agility, reduced reliance on legacy skill sets, and improved compliance.

Reduced Total Cost of Ownership

Historically, many organizations acquired “one of everything,” so their current IT environments are very expensive to maintain. Organizations already spend a large percentage of their IT budget on the maintenance of legacy applications—and this cost is increasing.

Today’s IT budgets are either frozen or decreasing. To free up funds to address ongoing business needs, IT organizations must reduce their budget spend on legacy applications and environments.

IT modernization reduces costs by

- Optimizing business processes to save labor costs
- Automating previously manual processes to further reduce the cost of labor
- Reducing the need to extend or modify applications through the use of reusable service-oriented architecture (SOA) components
- Reducing or eliminating ongoing maintenance and support fees for expensive, proprietary legacy infrastructure

Organizations can achieve these cost reductions by adhering to three key principles:

- Using lower-cost software platforms
- Leveraging packaged applications where possible
- Consolidating technology and technology providers
Using Lower-Cost Software Platforms

To reduce costs, organizations must use lower-cost computing platforms based on more-modern software technology—such as Oracle Database, Oracle Fusion Middleware, and Oracle Enterprise Manager—to form an application and database grid infrastructure that acts as an open, integrated, and highly scalable unit. In customer benchmarks and numerous production deployments, these systems deliver mainframe-class quality of service characteristics, often at a fraction of the cost. Grid-enabled environments range from mainframe emulation and best-of-breed hosting of COBOL/C logic to leading Java/J2EE application servers with seamless, end-to-end integration that can quickly deliver savings and enable long-term benefits.

In turn, grid computing platforms combined with SOA create the next-generation IT environment where orchestrated application components combined with computing resources in multiple locations form a virtual environment with a single point of management, control, and access. Oracle supports exactly such an architecture—an architecture that is product-independent and can be used with both Oracle and non-Oracle products.

Taking Advantage of Packaged Applications

Whenever possible, organizations should consider using packaged applications if the applications fit with the needs of their business. Packaged applications should adhere to SOA standards and work with the organization’s underlying architecture. Purchasing applications that cannot easily be integrated using SOA creates new application silos, keeps costs high, and decreases the agility needed to react to business change.

Oracle offers the most complete portfolio of business applications delivered on a secure and open technology infrastructure. Oracle Applications can support your business end-to-end and top-to-bottom with both line-of-business and industry-specific solutions ranging from finance to freight management to sales.

Combining packaged application solutions with SOA-enabled grid platforms for unique custom components enables the flexibility that IT needs to respond rapidly to business needs and competitive pressures. Using open standards, Oracle Application Integration Architecture serves as the foundation for deploying and integrating packaged applications alongside existing and new software applications. Oracle Application Integration Architecture enables a modernized enterprise to function cohesively and helps it meet new business requirements in a timely manner.

Consolidating Technologies and Providers

Using an assortment of highly proven IT modernization approaches, organizations can consolidate technologies and technology providers while determining the best combination of modernization approaches for each application. (The best combination of approaches for an application is based on the business needs of the organization.) For more information about the types of modernization approaches and when to use them, see the Oracle white paper, *Oracle IT Modernization Series: Approaches to IT Modernization.*
Increased Agility

When adapting to business needs, legacy applications are anything but agile. The architectural concepts underlying legacy applications do not reflect the way business works today—processes that are easy to change in business are often difficult and costly to change in legacy computer applications.

As IT organizations work to transform themselves for better alignment with business requirements, they are recognizing the need to move toward process-driven SOA. Process-driven SOA allows the use of individual application components as services—that is, the components are located and accessed only when needed at execution time. Creating services from legacy components can be difficult, but Oracle solutions can help IT organizations make timely, cost-effective transformations by using a variety of modernization approaches.

In addition, using SOA services in combination with process-orchestration engines capable of driving services (such as Oracle BPEL Process Manager) lets IT organizations create applications that more closely reflect the organization’s process flow and business procedures. Such process-driven applications are easier to enhance and maintain because process and workflow changes are elevated above individual services and are maintained in an easier-to-change orchestration layer that uses a declarative approach without affecting reusable SOA components.

Reduced Reliance on Legacy Skill Sets

People with skill sets in legacy technologies are getting harder and harder to find, creating an ongoing and ever-increasing risk for all organizations with legacy applications. Knowledge of languages, such as ADSO, NATURAL, or IDEAL, and expertise with databases, such as ADABAS, IDMS, and Datacom/DB, are increasingly scarce and expensive. People with skills in mainframe systems and application development (or maintenance) are also rare. Most programmers no longer learn legacy technologies in school and, even if they were trained in them, prefer to work in environments that support the latest technologies. Organizations need only check with their human resources department for the retirement dates of personnel with legacy skills to determine when the sustainability of legacy application systems will become a problem.

If the trend toward diminishing availability of legacy skill sets continues, many experts believe that IT organizations will begin to experience a legacy crisis—with fewer and fewer IT resources devoted to the development of new systems.

Improved Compliance

Government regulations such as Sarbanes-Oxley require that CEOs and CFOs verify that their systems are doing precisely what they claim they do. Executives cannot safely vouch for their systems unless the company fully documents its processes.

To improve transparency and track ongoing business process changes, many organizations are taking advantage of process-driven SOA to create applications that reflect and implement application processes as they are defined by the business. Using SOA components makes it much easier to track the current processes, as well as when and how they change.
Why Oracle Modernization?

Oracle offers an array of IT modernization approaches in support of the complex variety of legacy modernization challenges. These approaches include

- Replacing legacy applications with packaged, commercial off-the-shelf (COTS) applications
- Enabling service-oriented architecture (SOA)
- Rearchitecting legacy applications
- Automating the migration of legacy languages based on 4GLs and other legacy languages
- Offloading a mainframe’s MIPS transactions to an open system using data caching
- Rehosting application logic and data, intact, to a more-open, cost-effective, and agile platform

A combination of these approaches is typically required to provide a complete solution to the business problems. (See the Oracle white paper, Oracle IT Modernization Series: Approaches to IT Modernization, for details on these approaches.)

Oracle Delivers for Legacy-Software Customers

While some organizations delay because of risk concerns, Oracle customers have embarked on the modernization journey with confidence. Mainframe-based legacy application systems are well known for their high quality of service attributes, but the cost remains very high. The robust foundation provided by Oracle Database Grid and Oracle Application Grid (based on Oracle Tuxedo, Oracle Coherence, and Oracle WebLogic Server) and deployed in Oracle’s Maximum Availability ensures that the modernized application systems offer availability and performance that’s equal to or higher than mainframes. Oracle customers have experienced mainframe-class reliability and performance after migrating to open systems; they have also gained scalability and availability advantages at a fraction of the cost.

Reality provides the proof. Business systems based on Oracle technology have delivered greater than 99.999% availability for many customer production environments, including those that manage reservations, handle government benefit transactions, and deliver business-critical financial services such as mobile billing. Some of these applications run at tens of thousands of transactions per second (tps) in production; in some customer benchmarks, the Oracle infrastructure exceeded 100,000 tps while maintaining a subsecond response time.

To get the maximum strategic business benefit, it’s important to base the modernized system on an architecture built on open standards and deployed on open systems. Using Oracle’s complete, open, and integrated software solutions, organizations can implement business systems that align IT and business strategies and deliver a preintegrated solution at a lower cost—letting businesses save not only on the initial deployment and integration costs, but also on ongoing operational and maintenance expenses. Oracle’s complete, open, and integrated software enables top-line revenue growth while reducing bottom-line expenses, resulting in increased profit margins for the business.
IT modernization can be done as quickly or as slowly as an organization requires. Strategies and road maps can span multiple years, but must align with the organization’s business priorities and budget constraints. To determine the best IT modernization approach or combination of approaches for a specific organization, Oracle works closely with its modernization partners. Sometimes a customer may ask for a recommendation for a qualified partner, but many Oracle partners are already preferred vendors of our customers.

Planning an IT modernization effort includes defining a strategy and developing a plan that maps the current legacy environment to the desired state. The planning process also includes identifying a target architecture, acquiring the required software, and creating a multiphase execution plan.

Modernization projects typically require a combination of three solution providers: a systems integrator to manage the overall project, a modernization vendor to provide process expertise, and a technology vendor to provide the new target environment. As part of its comprehensive Oracle PartnerNetwork, Oracle has put together the Oracle Modernization Alliance: best-of-breed vendors to collaborate in a modernization partner team that can address any modernization project.
Conclusion

IT organizations are under increasing pressure to reduce costs and increase their ability to react to ongoing business demands. Legacy applications continue to be a problem for organizations because these applications are expensive and difficult to maintain and do not meet the needs of today’s businesses.

IT modernization is the continuous evolution of an organization’s existing applications and infrastructure software, with the goal of aligning IT with shifting business strategies. IT modernization implies the acquisition and deployment of modern technologies—along with their associated skill sets and capabilities—to replace legacy environments, without having to start from scratch or take the risk of rewriting business-critical applications.

Oracle’s open, complete, and integrated infrastructure and application software can be the foundation of an IT modernization effort that reduces total cost of ownership, increases agility, eliminates reliance on legacy skill sets, and improves compliance.
Highly efficient and competitive organizations understand that an effective IT environment is a major pillar of their success.

Unfortunately, IT environments are often weighed down with legacy components that are costly, risky, and slow. The resulting imbalance between IT and business creates an execution gap that keeps IT from successfully supporting ongoing business requirements—requirements that are being driven by customers, competitors, and demands for compliance.

The solution for this execution gap? IT modernization.

IT modernization is the continuous evolution of an organization’s existing application and infrastructure software, with the goal of aligning IT with ever-shifting business strategies. IT modernization implies the acquisition and deployment of modern technologies, skill sets, and capabilities to replace legacy environments. These modern technologies must be based on open standards and must provide an open, complete, and integrated environment that is both economically efficient and able to support an organization’s strategic business goals.

IT modernization can be done as quickly or as slowly as an organization requires: strategies and road maps can span multiple years, but must align with the organization’s business priorities and budget constraints.

This paper examines the various approaches to modernizing an IT environment. Typically, organizations will need a combination of approaches for a complete IT solution.

Planning an IT modernization effort includes defining a strategy and developing a plan that maps the current legacy environment to the desired state. The planning process also includes identifying target architecture, acquiring required software, and creating a multiphased execution plan.
Approaches to IT Modernization

IT modernization is the continuous evolution of an organization’s existing application and infrastructure software, with the goal of aligning IT with shifting business strategies. It implies the acquisition and deployment of modern technologies—along with their associated skill sets and capabilities—to replace legacy environments.

Oracle offers several approaches to IT modernization to address the complex challenges of modernizing legacy environments.

- Replacing legacy applications with packaged, commercial off-the-shelf (COTS) applications
- Enabling service-oriented architecture (SOA)
- Rearchitecting legacy applications
- Automating the migration of legacy applications based on 4GLs and other legacy languages
- Offloading a mainframe’s MIPS transactions to an open systems platform using data caching
- Rehosting application logic and data, intact, to a more-open, cost-effective, and agile platform

Organizations typically need a combination of these approaches for a complete modernization solution. Each of these approaches is examined in the following sections.

Replacing Legacy Applications with Packaged Applications

The approach that is most often considered in IT modernization is replacing legacy applications with one or more packaged COTS applications. This approach includes replacing horizontal, functional applications as well as vertical, industry-specific applications. Replacing legacy applications with SOA-based application packages can also be highly cost-effective.

Of course, legacy applications can be replaced with packaged applications only when appropriate application replacements exist. Both horizontal applications (such as payroll, accounting, billing, and customer relationship management) and many vertical applications (such as those designed for aerospace and defense, communications, industrial manufacturing, financial services, and utilities) are available today from Oracle and other software vendors.
To get maximum agility from replacement applications, organizations should replace legacy applications with applications made up of SOA components and applications that use SOA capabilities (such as SOA component orchestration). These SOA components can then be mixed with other modernized components resulting from rearchitecting, rehosting, and automatically migrating custom application components using an SOA platform and a services orchestration engine. This maximizes the agility of the complete application because the packaged applications are seen as sets of reusable components rather than as isolated application silos.

Even though packaged applications typically provide significantly more functionality than legacy implementations, there may be some unique facets of the legacy system that are not covered by the packaged application. In this case, the business logic from the legacy system can be retrained using one of the other modernization approaches described in this paper, such as rearchitecting or rehosting the application. The combination of Oracle Applications with Oracle Fusion Middleware using Oracle Application Integration Architecture allows for coexistence and integration of packaged and custom processes, giving customers greater flexibility.

Enabling Service-Oriented Architecture

This approach to IT modernization consists of wrapping legacy application services in place and presenting them as Web services to an enterprise service bus. The value of this approach is that it provides immediate integration of legacy systems with other systems, using middleware such as Oracle SOA Suite.

One of the key business benefits of enabling SOA integration is that organizations can buffer business processes from business applications so that changes to the application systems—or to the software they are built on—are not exposed to the process layer. The system is improved with limited impact—possibly no impact—on the business process.

SOA integration also allows organizations to move faster: They can run more projects concurrently because they know the projects will be tightly integrated at the end of the process. Registered SOA services can be reused and combined with applications that have been modernized through other, more-complex modernization techniques, such as rearchitecting.

By enabling SOA integration, an organization can begin to use SOA concepts—including the orchestration of SOA services into business processes—and leave legacy applications intact. Of course, appropriate interfaces to the legacy application must already exist, and the code behind these interfaces must perform useful functions that can be packaged as services.

SOA integration typically takes advantage of three legacy application interface points.

- **Presentation screens.** A legacy screen or group of screens is replaced with an SOA service that drives the underlying program the same way the original screen did. Presentation screens are often good candidates for SOA integration because many legacy applications use screens to drive a single transaction, such as adding a customer or approving a purchase order.

- **Functional calls.** Calls to a legacy procedure or program are replaced with an SOA service that issues the same call. Procedures or programs that can be called by the legacy application may have been written as reusable components and are candidates for reusable services.

- **Database calls.** A call to a legacy database or file system is replaced with an SOA service that issues the same native call and returns the requested data. Because the new environment uses relational databases for handling data, this call may be further modernized by allowing a structured query language (SQL) call to be issued, even though the data is not stored in a relational database environment.
SOA integration often involves communicating with the existing application on its current platform. However, it is also possible to use the SOA integration technique with an application after rehosting to another platform. (See “Rehosting Legacy Applications,” later in this white paper, for more information.) Using SOA integration in this way is useful because all the components—the rehosted application components that have been integrated with SOA, the new Java components, the packaged application components, and the orchestration engine that brings them all together—reside on the application grid leveraging common management, monitoring, and virtualization capabilities.

Because SOA integration does not typically require deep changes in the legacy application, legacy components can very quickly be used as part of an SOA infrastructure with little risk. Enabling SOA integration is definitely a first step toward a completely modern environment. However, because the code that implements the services remains unchanged, SOA integration does not solve the problems involved in maintaining a legacy environment, and provides only partial help in reducing the total cost of ownership (TCO), increasing application agility, reducing dependence on legacy skill sets, and achieving compliance.

Rearchitecting Legacy Applications

As an approach to IT modernization, rearchitecting means building new, replacement systems on the side; integrating these new systems with the old systems; and eventually, shutting down the old systems.

Legacy applications are a mix of business-relevant and technical code that implements legacy technical-support capabilities. Because the new IT environment provides much of this support functionality in other ways, the original technical code is no longer needed. Rearchitecting, then, focuses on recovering and reassembling business-relevant code from legacy applications while eliminating as much of the technology-specific code as possible.

Rearchitecting is typically used in modernization projects that involve changes in architecture, such as introducing object orientation and process-driven services. Rearchitecting recognizes that—in addition to the application code—application process interactions, data models, and workflows are also useful to the rearchitecting process.

Rearchitecting is typically done in four phases: recovery, redesign, refactoring, and regeneration.

Phase 1: Recovery

In the recovery phase, the original application is analyzed using an application portfolio analysis (APA)1 along with micro- and macro-analysis techniques to create a clear understanding of both the application and the models behind it. (These models are typically in the form of legacy modeling techniques, such as information engineering models.) This information is gathered top-down from walkthroughs of the existing system, existing workflows, and corporate process models. A bottom-up analysis is then done of existing data models, data dictionaries, and the legacy application code itself. The results of all these analyses are placed in a repository in preparation for the refactoring phase.

The recovery phase provides a platform-independent, present-case model of the current application design, as well as the analyzed code that implements the present-case model. The code may be represented either in an abstract syntax or in its original syntax. Even though the complete rearchitecting process requires human intervention, much of the recovery phase can be automated—especially when gathering information from existing computer sources.

---

1 An application portfolio analysis (APA) helps an organization better understand its current application environment.
Phase 2: Redesign

In the redesign phase, a future-case model—based on SOA principles and new modeling techniques—is developed to produce applications that are able to execute on SOA modernization architecture. New workflows are developed to cover the content of the present-case model, as well as the content obtained from use cases in the future-case model. New business process steps are developed using techniques that better align the new processes with the business processes and with the new, graphic system. Business-to-business interfaces are defined to cover the functionality of previous interfaces and to add new interfaces. This future-case model is stored in the same repository as the present-case model.

Redesign is often used in conjunction with the modernization approach of replacing legacy applications with packaged applications. If a packaged application exists that can replace all or part of the legacy application, then the packaged application components will form part of the redesigned architecture.

Components of the future-case model may correspond closely to the present-case model, but rearchitected changes or desired changes in business operations may mean that the two models differ substantially. For example, the redesign phase may examine processes to determine which are batch processes due to business requirements, and which are batch processes due to legacy technology constraints (such as CPU availability) that could be refactored. (See the next section for more information on refactoring.)

In some cases, design concepts that were coded in the legacy environment are replaced by native functionality in the new environment. For example, reports may be replaced by data warehouses and reporting tools, batch job control may be replaced with orchestration flow, data edits may be replaced with database stored procedures, and some business rules may be programmed into a business rules engine. This type of replacement mapping allows for a reduction in the number of lines of code in the new system, resulting in a more agile system that costs less to maintain.

With both the present-case model and the future-case model stored in the same repository, the stage is set for the refactoring phase.

Phase 3: Refactoring

In the refactoring phase, organizations examine the business logic of the present-case model to determine what can be mapped to the packaged application components that make up the new design architecture. In this way, rearchitecting provides a way to compare the mined legacy content to the new application functionality and ensures a more complete form of gap analysis than can be achieved by just talking to end users.

Functionality that does not map to packaged applications or to the native capabilities provided by the new technology environment is transformed and reassembled into new components that are based on the future-case model. For example, a set of individual procedural routines to add, update, approve, pay, or delete an invoice would become methods on an invoice object and would be used by a change-invoice process. The routines might be further refactored to eliminate and reconcile duplicate edits that were repeated in a number of places because of cut-and-paste development.

Because of the nature of refactoring, it can never be completely automated. However, it can be partially automated using tools that analyze legacy content to find candidates that map to the packaged application environment or candidates that can be reassembled into new components. For example, once a file is identified as a candidate for a new business object, code analysis can trace back from any file updates to determine the code that affects the field
values comprising the file records. The actions performed by the code then become candidates for packaged application capabilities or for methods on a new business object.

When complete, the refactoring phase yields a future-case application—defined at a platform-independent level and designed to maximize the use of packaged applications, SOA, and the capabilities of the new environment. Content that cannot be refactored to packaged applications can be refactored directly to a language, such as Java. Sometimes, a higher-level abstract representation is needed—for example, an application development framework that uses regeneration techniques.

Phase 4: Regeneration

The final phase of rearchitecting is regeneration. This phase maps nonpackaged applications, platform-independent refactored models, and code abstractions into a platform-dependent form. To carry out this mapping, the regeneration process takes advantage of application development frameworks and integrated development environments. These frameworks and tools make the creation of the final application simpler by providing both design templates and reusable implementation components that make maximum use of the new environment. At the same time, these frameworks and tools help hide any remaining platform specifics from the application code itself—increasing flexibility and lowering the cost of maintenance.

Although regeneration can be a manual process, in most cases application frameworks and design patterns make automation possible. Regeneration techniques can also be used to incorporate specific platform requirements, such as Java technical architectures.

Because rearchitecting takes maximum advantage of the knowledge contained in the existing application, it costs less and has fewer risks than developing an application from scratch. However, because rearchitecting does involve a high degree of change, there are still risks.

Other modernization techniques have been developed that do not offer all the benefits of rearchitecting. These other techniques can be used to break the process of modernization into a series of steps, which—although more costly—may have fewer risks.

Automating the Migration of Legacy Applications

In some cases, the process of migrating legacy applications to a new architecture can be automated. A migration is considered to be automated if at least 80 percent of the migration or transformation can be handled by migration tools rather than manually. Depending on the nature of the technologies involved (languages, databases, and so on) the degree of automation can range from 50 percent to the high-90s. To reach this degree of automation, the migration or transformation process must be algorithmic and should not require human intelligence during the transformation process (as is needed, for example, in the rearchitecting processes).

Automated migration does not typically change the design of an application, but it can provide specific enhancements if they are incorporated in the set of rules driving the tools (for example, field extensions). Automated migration takes existing code and runs it through a parser to create an abstract representation which is then fed to a utility that generates code in a new language (for example, migrating COBOL or Natural to Java). Automating migration is fast, but it only works if the gap between the legacy architecture and the new architecture is relatively small. Automated migrations are most successful when there is a well-defined mapping between the source and target architectures. Mapping that is not well-defined will cause problems with the migration, as described in the following sections.
Transforming Procedural Designs into Object-Oriented Designs

Although it is possible to transform procedural code (such as COBOL or PL/I) into an object-oriented language (such as Java), it is not possible to map the procedural design that surrounds COBOL programs into the object-oriented design that surrounds good Java programming. The fundamental design concepts of object-oriented programming—such as the class and its behavior—are architectural concepts that require human intelligence to design. The designer of a truly object-oriented application will use these techniques in new ways that cannot be recovered from an application developed using nonobject-oriented techniques. The challenges involved in maintaining applications that have been automatically migrated to a “procedural Java” implementation can be significant because there is no easy way to extend these applications using standard Java methods, class-based techniques, and so on.

Migrating legacy applications to object-oriented designs cannot be successfully automated using our definition of automated migration (where at least 80 percent of the migration is done with technology). This is equally true when automatically transforming legacy procedural flows into more-modern modeling approaches such as a Unified Modeling Language (UML). The transformation can be done, but the resulting UML will be a present-case UML—not the same UML that would be designed for a workflow-driven application.

Transforming Pseudo-Conversational Code into Conversational Code

One of the most interesting concepts in the history of computing was the introduction of pseudo-conversational programming by IBM in the 1960s. This concept, introduced at a time when machines were expensive and machine space was at a premium, forced programs to drop context information—including variable content and transactional content—from each screen input or output.

Although the loss of context with pseudo-conversational code is similar to the loss of context caused by thin client HTML applications, pseudo-conversational code is not the same as HTML when it comes to transaction handling—and it is very different from a conversational application where the application context is retained across any application display or event. Pseudo-conversational code requires human intelligence to map a set of pseudo-conversational screens to conversational code; transforming pseudo-conversational code to conversational code, then, cannot be successfully automated.

Pseudo-conversational code was developed and implemented to drive the efficiency of resource consumption, enabling tremendous scalability. While current-generation software employs different methods to achieve similar goals, if the resulting systems are to be as efficient and scalable as pseudo-conversational code, any transformation should be considered thoughtfully, with great care and attention to detail.

Transforming Legacy Procedures to Business Services

Like the SOA integration and rehosting approaches, automating migration does not change the core structure of the legacy application. To use migrated code as services, the legacy interfaces must first exist. Many mainframe applications developed on IBM CICS or IMS TM are structured in a way that enables reasonable mapping of business transactions to fine-grained services. Screen-based applications often offer basic services (for example, creating an invoice, updating an invoice, recording the payment of an invoice, and so on) which can be mapped to fine-grained services in a SOA. However, in some cases, the ideal services may not exist in legacy applications because the legacy applications simply were not designed that way.

Technical SOA interfaces can be created as part of an automated migration; however, business services that more closely follow the business process are much more difficult to create automatically unless the legacy components needed to support the services already exist.
Transforming to a Business-Process-Driven Environment

To get the maximum benefit from a new IT environment, the modernization process must transform legacy applications into SOA services. In this case, a business process automation tool is used to define both the human and computer processes that are driven by a business process management engine.

Unfortunately, most traditional applications have either no workflow concept outside of batch control scripts, or the workflow is embedded deep within the code. Transforming these applications so that the workflow is at the top of the architectural stack and can drive the rest of the application is not something that can be successfully automated. In fact, it may be that determining the current workflow will require interviewing current users—something that can’t be automated at all. In most cases, creating applications that use orchestrated SOA services to align with business processes cannot be done automatically.

Transforming from Batch to Online Processing

In many legacy applications, some aspects of online transactions are recorded and then processed later using batch processing. For example, stock trades cannot be valued until the stock market closes, so final processing cannot occur until the end of the day. Batch processing, such as end-of-day reconciliation or end-of-month reporting, is based on human clock cycles and does not disappear in the new IT environment. Some transactions will continue to be processed this way.

However, there are many traditional batch processes that can be wholly or partially modernized and transformed into online processes. For example, incoming transactions can be processed immediately, resulting in more-up-to-date information. Transforming these batch processes to the correct forms of online processing requires human thought and cannot be automated.

Making Automated Migration a Viable Alternative

The main concern of any automated migration process—in addition to architectural changes—is the quality of the source code. If the source code is poor, transforming it automatically to another language or environment will not improve its quality. However, if the source and target design are similar enough, automated migration can be a viable alternative, producing fast, consistent results. In many cases, the issues making automated migration difficult can be avoided or even eliminated. These issues are discussed in the following sections.

Migrating Applications that Use Legacy Databases and File Systems to Relational Databases

Legacy applications that are written in third-generation languages (3GL), such as COBOL and PL/1, and that use legacy database and file systems (such as VSAM, IMS DB, ADABAS, IDMS, and Datacom/DB) retrieve data by issuing calls embedded within the program code. Once a data model maps the legacy database or file formats to relational tables, you can use automated migration to remove the calls and replace them with SQL calls. The rest of the code is not affected. The same data model mapping can be used to migrate actual data from the legacy database or file system to a relational database.

Migrating Applications Written in Legacy Languages to Other 3GLs

While the vast majority of mainframe applications are written in COBOL, some mainframe systems still have programs written in IBM Assembler, PL/I, and other languages that cannot be well supported on open systems. Automated migration can be used to convert these components into more-acceptable 3GL alternatives, such as COBOL or C, for rehosting on open systems. For assembler migration, a multiphase approach is typically required to isolate business logic from technical use of assembler. Subsequently, the business logic can be
mapped to a COBOL or C representation with a good degree of accuracy. Customers with significant PL/I assets have also experienced success using automated migration that maps PL/I to C or to procedural Java.

Migrating Applications Written in Fourth-Generation Languages to Java or Rehosted COBOL

Fourth-generation language (4GL) environments (such as NATURAL, IDEAL, ADSO, and PowerBuilder) typically consist of both a language and a runtime environment that provide additional capabilities to support the executing programs. The 4GLs were created to move away from the complexity of 3GLs and environments by using a runtime environment to hide lower-level implementation details and provide a higher-level, more abstract interface for application developers.

As a result, 4GL environments are architecturally more modern than 3GL environments, and do not have as many of the architectural issues that make automated migration to Java virtually impossible for 3GL environments.

For example, an organization can automatically migrate a NATURAL/ADABAS application to Java/Oracle. The resulting application will not have a completely object-oriented design because the original NATURAL application was procedural. However, because NATURAL applications are coded at an abstract level that uses a conversational programming style, the procedural Java code created will be conversational and can use object-oriented libraries that supply the needed environmental functionality.

The result of this type of migration is a mixed-mode application that can take advantage of new capabilities and is reasonable to maintain. The code can be modernized even further to get all the benefits of the new environment.

Organizations can automatically migrate legacy mainframe 4GL languages to COBOL and then rehost them. Although this may seem like a step backward, the fact that 4GLs are generally procedural and use mainframe concepts (such as mainframe data typing) means that migrating them to COBOL is relatively straightforward. As part of the migration to COBOL, legacy databases associated with 4GLs can be eliminated, and the data can be rehosted on a relational database. This technique is particularly useful when a legacy 4GL application is intertwined with legacy COBOL.

Restructuring Program Code

To prepare code for other types of modernization, you can use automated migration techniques to "clean up" the code. Program and loop restructuring can also be used to eliminate dead code and GOTO statements.

Automating Migration: Advantages and Disadvantages

The clear advantages of automated migration are speed and consistency. Because a computer carries out the modernization process, it can be done quickly and consistently, and can even be repeated on a more recent copy of the source code to include ongoing changes. Modernization is done the same way every time, so even though automated migration is more invasive than either SOA integration or rehosting, it has a lower risk and requires less testing than a manual effort.

The disadvantage of automated migration is that only algorithmic transformations can be made. If the goal of a modernization effort is to make major changes to architecture and application design, automated migration will not work.
Offloading MIPS

Using a data grid to offload MIPS via data caching is an approach to legacy modernization that uses an open-system–based, in-memory data grid solution to alleviate excessive and unnecessary traffic between midtier servers and a mainframe. Applications cache data in the data grid and avoid expensive requests to back-end mainframe data sources. Reading from the cache is faster than querying the back-end mainframe and scales naturally with the application tier.

Offloading MIPS using a data grid to cache the mainframe data is most successful when extensive Web traffic hits the mainframe and relies on fairly static data, as with daily rate tables, account balances, and so on. Creating J2EE services around the data grid helps maintain the currency of the data, but if these services connect to the mainframe for updates as frequently as the original traffic, little time would be saved.

Oracle Coherence is the industry-leading in-memory data grid solution for legacy modernization. It alleviates latency problems and drives dramatic increases in application performance by moving data closer to SOA applications for efficient access. In-memory performance alleviates bottlenecks, reduces data contention, and improves the application’s overall responsiveness.

In some cases, customers have made the data grid the master repository of such data, leveraging its scalability and cost-effective virtualization technologies. When mainframe services require access to this data, they leverage mainframe integration technologies, such as Oracle Tuxedo Mainframe Adapters, to access that data in the grid and in trigger-related J2EE services.

A key business benefit of implementing a MIPS offload solution is the significant reduction of MIPS usage and the associated mainframe costs. In many cases, the MIPS offload approach serves as the first step in strategic, multiphased IT modernization projects. Money saved from the reduced MIPS consumption is then invested in subsequent modernization activities that offer more-substantial, longer-term business benefits.

Rehosting Legacy Applications

According to some industry estimates, more than 200 billion lines of COBOL code exists in enterprise applications—mostly in mainframe environments—and this number is growing by 3 to 4 percent a year. Rehosting is a modernization approach that is predominantly focused on migrating mainframe COBOL—intact and with no application logic changes—to open, lower-cost platforms. Of course, many mainframe applications rely on additional technologies, such as transaction monitors (such as IBM CICS, IMS TM, or Unisys MCS), pre-relational and relational databases (including IMS, IDMS, and DB2), and record-oriented file systems (such as VSAM, 3270, and terminal emulation). In addition, many applications have batch components that run in a job entry subsystem 2 or 3 (JES2/JES3) environment, orchestrated by job control language (JCL) and controlled by schedulers, such as CA-7 Workload Automation (CA-7), CA-11 Workload Automation Restart and Tracking (CA-11), BMC CONTROL-M for Distributed Systems, and so on). Rehosting has evolved as a complete methodology to address all of these mainframe technologies surrounding online and batch applications, leveraging a combination of emulation solutions (where possible) and automated migration (where required, for example in migrating a 4GL to COBOL and then rehosting it off the mainframe, or in migrating PL/I to C before rehosting it to open system platform).

As a proven and low-risk approach requiring minimal change, rehosting is best used when most of the application meets business requirements and quick migration can provide significant cost savings by eliminating, reducing, or containing MIPS consumption on the
mainframe. When migrating a small-to-medium mainframe environment, rehosting has been successful in significantly reducing the use and cost of the mainframe, and sometimes in eliminating the mainframe dependency altogether. For larger mainframe shops, rehosting for MIPS containment provides enough breathing room to avoid—or at least defer—an expensive mainframe upgrade.

The rehosting approach is based on

- Migrating the application off the mainframe to a compatible software environment on an open systems platform
- Preserving the language (if it is COBOL or C) and middleware services on which the application has been built

Rehosting protects legacy investments by relying on a mainframe-compatible software stack to minimize any changes in the core application and preserving the application’s business logic while running it on an open system operating system (OS) that has a more flexible and less expensive system infrastructure. Rehosting also keeps the customer’s options open for SOA enablement and rearchitecture by using a SOA-ready middleware stack to support Web services and enterprise service bus (ESB) interfaces for rehosted components. By using an extensible software platform that is open to integration with Java, J2EE, or .Net components; has BPM-based processes; and uses other key tools of the rearchitecture approach, customers can start rearchitecting selected components to support targeted functional changes at any time and still maintain strong integration with the rehosted services that are running the bulk of the business logic.

Reducing or eliminating the legacy mainframe costs and risks via rehosting can also help customers fund subsequent SOA enablement and rearchitecture aspects of legacy modernization and lay the groundwork for these approaches. Enabling SOA for a rehosted application is a much easier process on an open-systems–based, SOA-ready software platform. It is more efficient in its use of resources utilization, and therefore less expensive.

Rearchitecting selected components of a rehosted application is a lower-risk approach than rearchitecting the entire application. The risk can be further reduced by making sure that the target rehosting stack provides rugged, transparent integration between the rehosted services and the new components.

Three key success factors emerge from numerous mainframe rehosting projects:

- Leverage automation and proven migration practices to reduce project risk and its duration and to ensure a predictable schedule for cost reduction and demonstrable business value
- Preserve the application’s business content—both the application logic and the application data—to achieve functional equivalence sooner, minimize business impact, and avoid retraining
- Architect on an open system target environment to ensure mainframe-class reliability, availability, scalability, and other quality of service requirements and to ensure that migrated applications continue to meet performance and availability service level agreements

Meeting these needs requires a uniquely powerful database and application platform—one that natively supports key mainframe languages in a mainframe-compatible online and batch runtime; enables automated migration of application code and data; and delivers proven, mainframe-like quality of service on open systems. This database and application platform must also provide added flexibility so rehosted applications can be rapidly integrated in a SOA, and must be able to extend those applications with Java or .Net components as the needs of the business evolve.
Oracle’s rehosting solution combines Oracle Tuxedo (as the mainframe-compatible environment for hosting COBOL or C business logic) and Oracle Database. This mainframe-class platform natively supports COBOL and provides an execution environment compatible with the mainframe’s IBM CICS or IMS TM for online applications and provides a JES for batch. (See the Oracle white paper, *Mainframe Rehosting with Oracle Tuxedo: Accelerating Cost Reduction and Application Modernization*, for details.)

The robust foundation provided by Oracle Database and Oracle Tuxedo deployed in Oracle’s Maximum Availability Architecture on Oracle software technology ensures availability and performance that’s equal to or better than on a mainframe, and provides significant scalability advantages at a fraction of the cost.

During the migration process, rehosting carries a lower risk than attempting to migrate a large and complex application to a different language or architecture. Rehosting is an optimal choice when you need to preserve the core business logic and data of the mainframe application because the company continues to rely on the business processes supported by the mainframe. Rehosting eliminates the need to maintain antiquated skills for mainframe maintenance and can modernize the application platform and interfaces, making it easier to extend functions and integrate with other systems.

Rehosting can result in significant cost savings. While these savings can be achieved relatively quickly, the rehosting approach (as with the enabling SOA integration and the offloading MIPS approaches) retains some legacy architecture and procedural programming languages, which can present ongoing business risks because of increasingly rare legacy skill sets and potentially cumbersome implementation of services.

With a focus on deriving the maximum strategic business value while mitigating the risks presented by retaining legacy code, the rehosting approach is typically employed as the first step in strategic, multiphased IT modernization projects and can be followed by SOA enablement, partial rearchitecture, and other modernization steps. Budget savings from the rehosted applications can be invested in these subsequent modernization activities that offer additional business benefits. Rehosting can also be used to close the gap between the custom legacy application set and a packaged solution by providing a path forward for the existing functionality not supported by COTS applications.
## IT Modernization Approaches: Advantages and Disadvantages

<table>
<thead>
<tr>
<th>APPROACH</th>
<th>DESCRIPTION</th>
<th>ADVANTAGES AND DISADVANTAGES</th>
</tr>
</thead>
</table>
| **Replacing legacy applications with packaged applications** | This approach to IT modernization is considered most often. Replacing legacy applications with Oracle Applications could include horizontal applications—such as Oracle E-Business Suite, Oracle's Siebel Customer Relationship Management (CRM), and Oracle's PeopleSoft Human Capital Management—along with industry-specific applications.                                                                                                           | **Advantages:** Highly cost-effective.  
**Disadvantages:** Not possible with home-grown applications that have unique features and are critical to the business.                                                                                                                                                                                                                                                                     |
| **Enabling service-oriented architecture**     | This approach consists of wrapping legacy application services in place and presenting them as Web services to an enterprise service bus. It provides immediate legacy integration to Oracle SOA Suite.                                                                                                                                                                                                                                           | **Advantages:** Noninvasive to the legacy application. Legacy components can be used quickly as part of the SOA infrastructure, with very little risk.  
**Disadvantages:** The legacy code that implements SOA services remains unchanged, so the problem of maintaining a legacy environment remains.                                                                                                                                                                                                                                  |
| **Rearchitecting legacy applications**         | This approach builds a new system on the side to replace the legacy system, integrates it with the old system, and then eventually shuts down the old system.                                                                                                                                                                                                                                                                                                                                                       | **Advantages:** Maximizes the benefits of SOA and new technology capabilities.  
**Disadvantages:** Very costly.                                                                                                                                                                                                                                                                                                                                                                                                 |
| **Automating the migration of legacy applications** | This approach runs existing code through a utility to generate code in a new language (for example, from COBOL to Java).                                                                                                                                                                                                                                                                                                                                                     | **Advantages:** Speed and consistency.  
**Disadvantages:** Makes only algorithmic transformations. Not the right choice when the goal is to make major changes to architecture and application design.                                                                                                                                                                                                                             |
| **Offloading MIPS**                           | This approach offloads expensive mainframe MIPS to a mid-tier server through the use of an in-memory data grid (Oracle Coherence).                                                                                                                                                                                                                                                                                                                                                   | **Advantages:** Reduced mainframe utilization and costs and increased application performance.  
**Disadvantages:** Retains all of the legacy architecture and programming languages, forcing a continued reliance on legacy skill sets.                                                                                                                                                                                                                                                       |
| **Rehosting**                                 | This approach to IT modernization migrates an application as-is to another platform, while leaving the core application essentially untouched.                                                                                                                                                                                                                                                                                                                                                             | **Advantages:** Moves an application to another platform without changing the core application.  
**Disadvantages:** Retains much of the legacy architecture and programming languages, forcing a continued reliance on legacy skill sets.                                                                                                                                                                                                                                                     |
Conclusion

All organizations want to realize the highest business value possible from their existing investments in IT infrastructure and application software. However, maintaining the application and infrastructure software in a legacy environment consumes a disproportionate percentage of IT budget and human resources. The average company spends from 60 to 85 percent of its IT budget maintaining legacy applications that fail to meet the changing competitive needs of the business.

Oracle IT modernization approaches, as detailed in this white paper, support the complex variety of legacy modernization challenges and provide an IT environment that is both economically efficient and able to support an organization’s strategic business goals.

Using Oracle’s complete, open, and integrated software applications, organizations can implement business systems that align IT with business strategies and can grow the organization’s top and bottom lines.
Oracle IT Modernization Series: Why Modernize?

Learn More
Go to: oracle.com/goto/modernization