white paper

Rocket

Modernizing IBM i Applications

For Multi-platform Environments

Modernizing IBM i Applications for Multi-Platform Development

A White Paper by Rocket Software

Version 3.0 August 2014

To exist in today's multi-faceted,exciting and fast-paced world of business, a company must re-orient its IT effort to incorporate multiple platforms for diverse software applications. Building a multi-platform development process around the IBM i platform represents a unique opportunity for IT to combine the platform's traditional strengths of reliability, security, development productivity, ease-of-use, and management with new technological advances including WebSphere, Java, Linux, SOA, and much more. This white paper examines how organizations can take advantage of a comprehensive application lifecycle management system that addresses the added intricacies of multi-platform development.

Modernizing IBM i Applications

Introduction

Savvy companies today understand the potential of sophisticated new business applications. End users, customers, vendors and employees demand intuitive graphical user interfaces (GUIs), easy- to-navigate websites, automated business-to-business transaction processing and on-demand 24x7 data processing abilities.

IT departments around the world and in every industry are facing the same challenge—creating high-performance, secure and reliable software applications that span multiple platforms to support their business needs. IT must build applications that integrate components from distributed systems, Web-based systems and Windows.

Yet companies cannot afford to throw away years of investment by simply casting off legacy applications. Simply adding fancy GUIs without sufficient back-end support is ineffective and short-sighted. For companies that have already invested heavily in IBM i applications, the strategic advantage lies in modernizing and renewing the investments they have already made.

By successfully modernizing traditional business applications rather than rebuilding from scratch, companies can achieve higher productivity, higher return on investment and faster time to market. Application modernization is the answer.

IT in the Business World

The revolution of graphical user interfaces, pervasive communications, instant-access technologies (client/server and Web-based) and Service-Oriented Architecture (SOA) has succeeded. Today's customers and users demand nothing less than:

- Support for automated processing of information
- 24x7 online access to accurate product and company information
- Intuitive web-based ordering capability
- Graphical interfaces for internal applications
- Company-wide intranets

Every aspect of a company's operation must be modernized. Today's applications must include graphical order-entry processes, real-time inventory management systems, instant purchasing processes, and automated shipping programs—all available via the Web, 24x7, in multiple languages and customized for multiple sites. And the products must be easy-to-learn, easy-to-use, always available and always correct. Clearly, today's business climate is more exacting and less tolerant than ever. The complexity of these new applications makes failures more likely, while the accessibility of the applications makes those failures significantly more visible and costly.

Providing sophisticated business services requires a complex IT structure. No single server, language or platform is capable of answering every need. To be effective, IT organizations must incorporate multiple servers, build applications using new languages and support multiple programming platforms.

Application Modernization: The Crucial Nexus for Business Success

Billions of lines of code have been written on the IBM i platform. The resulting applications run crucial operations, from finance to manufacturing to customer relationship management. Now, business requires a new level of interaction for all applications. Rewriting all of this code would be costly, time-consuming and, in the final analysis, unnecessary. These applications simply need new interfaces and new ways to acquire and assimilate data.

Re-engineering applications may include:

- Implementing SOA initiatives
- Adding real-time components
- Creating graphical user interfaces
- Integrating middleware for cross-application and cross-platform communication

- Teaming IBM i servers with other systems
- Coordinating diverse development teams using multiple languages and multiple platforms in varying geographic locations
- Incorporating Web content with traditional code

What if there was a product that modernized applications to provide graphical interfaces, high availability and Web enablement? An IT manager could maintain the same level of high-quality software, continue to use the same data repository and enjoy the consistent stability and reliability of the IBM i server, while moving quickly to develop today's service-oriented business applications.

Such applications exist. Using IBM i applications on the market today, IT shops can create a graphical interface for an in-house inventory management application and create HTML code for an external Web page on top of existing applications without losing the underlying data, code, and content. The savings in capital expenditures alone from simply enhancing existing software and hardware make this an attractive proposition for any IT shop.

Taking this path to provide business services reduces initial investment cost and the expense of retraining, rehiring, and retooling and increases productivity by combining existing skill sets with new advanced technologies.

For IT managers addressing the new requirements for business success, building on IBM i applications and using the IBM i as a business server presents an ideal opportunity.

Multiple Platform Development Strategy

Managing application development in this new environment presents unique challenges for IT directors. With each new factor that is added to the IT mix, the complexity increases. In a multi- platform, multi-team business development environment, IT must address:

- Incompatibility of IBM i file storage technology with other distributed systems
- Inconsistent development processes and terminology
- Complex and variable build management processes
- Multi-platform code deployment
- Parallel development management
- Multiple development languages
- A wide variety of development tools
- Geographically dispersed and departmentally distinct development teams

In order to take advantage of the business opportunities inherent in multi-platform development, IT must establish a unified, structured, measurable, repeatable process capable of managing every language, every platform, and each development methodology. In short, IT must implement a multi- platform development process. A comprehensive Application Lifecycle Management (ALM) solution is the answer.

However, not all ALM solutions can address this range of issues. Our unique, proven approach to the application of ALM addresses each and every one.

Our single point of control over every stage of application development simplifies software maintenance and creation throughout the lifecycle. Further, our focus on inventory management and process automation is unique in the industry. We produce high-quality business applications efficiently and effectively by:

Managing the Inventory

In the multi-platform environment, IT must keep track of every type of file and object, including RPG, Cobol, CL, DDS and other standard IBM i objects, as well as distributed systems, Windows, and Web files. IBM i, distributed systems and Windows files are each stored differently, employ different naming rules, and reside on varying machines.

When developing applications across platforms, IT must ensure that all of these files and objects are maintained accurately with a detailed inventory of all of the parts that make up an application.

Our software automatically builds, organizes, and maintains a central inventory of all application components, ensuring that logical structures tie directly to business requirements. Our solution provides descriptive information, the storage location of every copy of each part, the current stage within the development process containing each copy, and which version of a part belongs to which release. It also tracks parts that are related. As code moves through the development process, the inventory system dynamically updates itself. Further, the system ensures that users know the status and location of every copy of every application part at all times.

Our inventory management system also operates as a services metadata registry. Optimized for SOA, the registry is easy to navigate and instantly accesses all services.

Automating Processes

In addition to significantly reducing administrative overhead, automating the move-to-production process reduces the number of errors in production. Move-to-production automation ensures that users know how every part in production got there and that every part is tested.

In a multi-platform environment, each platform and every development language maintains distinct development attributes and processes; therefore, process automation is more complex. Files must be accessed, moved, and compiled in the manner appropriate to their target platforms.

In an integrated application environment, many files are dependent upon each other. It is critical that changes made to one set of files be synchronized with corresponding changes in another set.

Rocket Aldon Application Lifecycle Manager (LM) automates and manages business processes through simple point-and-click operations. Users define a portfolio of processes to meet their application development requirements. The system automates and enforces these processes, minimizing risk to the organization, reducing production cost, increasing productivity, and simplifying regulatory compliance.

Automating Deployment

In a traditional development lifecycle, code is deployed to a variety of machines. In a multi-platform environment, the code is deployed not only to multiple machines but also to varying operating systems. As code is moved into production, the IT process must ensure that all of the files go to the right directories or libraries on the right platforms. The system must identify and match which code goes to what machine and, more importantly, it must be able to get the code there automatically.

Our powerful ALM solution automatically deploys all necessary components to the appropriate target locations throughout the development lifecycle within multi-platform, distributed environments. At each lifecycle stage, the automated deployment process gathers, packages, distributes, and installs application components. Through a management console, users easily track and manage the distributed packages.

Establishing and Controlling the Release Process Supporting the Developer

In a multi-platform environment, IT can eliminate many problems by implementing a formal release process. It is no longer prudent to maintain a process in which developers frequently make changes and move them to production individually. The number of variables makes an informal process a development hazard.

Developing by release allows IT to aggregate groups of related changes together. Managing by release ensures that changes are designed with common goals, reduces the amount of re-work associated with each change, allows testers to perform more comprehensive system tests, and makes it possible to more accurately track delivery performance—ultimately increasing both productivity and predictability.

To manage by release in a multiple platform environment, our software handles multiple paths to production, supports isolated testing environments for each release, manages file and object versions to provide developers with the right version from the appropriate release, and ensures conflicts between changes in different versions are solved.

The ultimate benefit of a multi-platform development environment is allowing users to choose the best environment for each problem. For example, just as the IBM i is the appropriate choice in terms of server stability and reliability, a UNIX machine can serve as the network's most efficient edge-server, while a Windows client can provide the graphical interface. Developers work most effectively in the environment in which they are most comfortable.

An IT development process within a multiple platform environment must allow developers to work within their own IDEs, allowing transparency by creating a common interface to the development system. Our software allows developers to work within their preferred IDE—Windows, Eclipse, RAD, WDSc or green screen.

Providing a Common Interface

The development process itself must be accessible through a common interface. With multiple platforms to manage, it is critical that IT have a global view that identifies and locates all of the files within the development lifecycle regardless of the machine on which the code resides.

IT organizations have an ever-increasing variety of tools and methodologies to choose from for creating enterprise applications. This can greatly add to the complexity of managing the development process. Through the ongoing support of technical standards (e.g. Eclipse, SCCI, etc.), we provide the flexibility to choose the right tool for your IT organization.

Managing Tasks and Facilitating Collaboration

As IT implements multiple platforms and languages, enabling effective communication among an increasingly varied group of people is a challenge. Today's business applications integrate the needs of many different kinds of end-users and require the skills of a wide variety of developers. To overcome cultural boundaries, Rocket Aldon LM provides a comprehensive collaboration and task management system, enabling developers and users to communicate with each other and stay abreast of changes that affect and/or involve them.

Providing Metrics, Dashboards and Reporting Capabilities

We provide metrics, dashboards and reporting functions to quickly access real-time information on productivity, staff utilization, application development progress and incident resolution statistics and other historical data to improve and optimize the development process. Customizable reports and dashboards offer instant overviews of project status, metrics and key performance indicators, giving management the visibility and information needed to appropriately deploy resources and assess the return of IT projects.

Conclusion

To exist in today's multi-faceted, exciting and fast-paced world of business, a company must re-orient its IT effort to incorporate multiple platforms for diverse software applications. Building a multi- platform development process around the IBM i platform represents a unique opportunity for IT to combine the platform's traditional strengths of reliability, security, development productivity, ease-of-use and management with new technological advances including WebSphere, Java, Linux, SOA, and much more.

To help companies succeed in this new world, we offer a comprehensive multi-platform application lifecycle management system that addresses the added intricacies of multi-platform development, allowing management to take advantage of the unique opportunities that this new environment offers.

We address the enterprise-wide needs of IT, giving companies the foundation to manage defined, repeatable, automated, measurable, and traceable processes for every stage of the application development lifecycle. We are unique in our ability to provide the development flexibility, management and control essential for successful business application development.

white paper

Rocket

www.rocketsoftware.com
info@rocketsoftware.com