SOA solutions
White paper

IBM System z tools for CICS SOA development.

By Michelle Cordes, IBM Rational System z Ecosystem team, mcordes@us.ibm.com; and John Knutson, IBM CICS Tools and PD Tools Marketing Manager, knutson@uk.ibm.com

October 2007

Contents

2 Introduction
2 SOA application development–business pressures and challenges
4 Processes and tools for SOA development
18 Using IBM System z tools to develop SOA applications–a scenario
34 Conclusion
34 For more information

Introduction

The realities of the twenty-first century exert conflicting pressures on organizations that build and deliver IT services. The lines of business say “more, more, more,” while financial management says “less, less, less.” This paper focuses on the challenges faced by the development and testing teams that are responsible for creating the new applications demanded by their business users. It will look at how a combination of formal processes and IBM System z™ tools can help the development of a service oriented architecture (SOA).

The paper first outlines the business challenges faced by the IT development-management team and then shows how a formal process combined with some of the System z products can help to address them. A development life cycle is then played out following IBM Rational® Unified Process, showing the value provided by some of those products used to develop and test a new application.

A companion white paper discussed how the IT service-management team at JK Enterprises used a similar approach to manage this newly developed application.[1]

SOA application development–business pressures and challenges

The late twentieth and early twenty-first centuries have seen phenomenal growth in the application of IT technology, allowing businesses to grow and change as never before. Demand for more IT is increasing, driven by mergers and acquisitions, innovative products and new markets. The much-promoted vision that SOA leads to business flexibility is increasing the demand for even more change as chief executive officers and line-of-business management see SOA as an attractive route to competitive advantage.

SOA does benefit chief information officers (CIOs) by simplifying the IT infrastructure, converting fixed-function programs into reusable services and moving control logic out of hardwired applications and into easily configurable business process management (BPM) systems. Such benefits often come with a price in the form of a short-term increase in complexity as the new SOA infrastructure and application components are implemented alongside those of heritage applications. SOA also increases the pace of change, requiring more from already overstretched IT team members, who are responsible for testing and deploying changes to systems, and finding and fixing problems that arise. The team members who are in most demand are often those with IBM CICS®, IBM IMS™, IBM DB2® and other System z skills, because these platforms are frequently used as the enterprise *SOA hub* and so are the focus for more development activity than in recent years.

Chief financial officers (CFOs), on the other hand, see things differently. From their perspective, cost reduction is often the only way they and their financial teams can contribute to enterprise competitiveness, and so they cut budgets to encourage their CIOs to reduce the total cost of ownership of enterprise IT systems, which is often achieved by reducing staffing levels.

There is also a risk that the increased level of activity and reduced staffing will result in a decline in quality with new releases being deployed without proper care. The almost inevitable consequence is application failure, which can result in loss of customer confidence, lost business, dropping stock prices and even, in extreme cases, total business failure. Recent regulatory compliance legislation, such as the US Sarbanes-Oxley Act (SOX), and industry initiatives, such as the Health Insurance Portability and Accountability Act (HIPAA) and BASEL II, can impose severe penalties on companies that fail to exert adequate controls over the business of IT, as well as on their IT executives.

“Do more, consume less, be compliant – or else.” How CIOs and their teams can satisfy all of these demands simultaneously is the subject of this white paper.

Processes and tools for SOA development

Let's look at a scenario to help illustrate processes and tools for SOA development. JK Enterprises is a fictitious supplier to retail, small business and worldwide corporate customers that is used elsewhere in IBM SOA-related scenarios. Like many others in his position, the development manager for JK Enterprises must bring in new applications, but he must also come in under budget. In the past, the company used the traditional waterfall method of developing applications, and that proved to be quite successful. He realizes that this method of development will not allow his development group to respond to ever-changing business requirements as quickly as needed.

The development manager decides to implement an iterative development methodology with IBM Rational Unified Process® IBM (RUP®). This methodology will enable requirements to change throughout the process, which is important, because this new SOA project is a totally new objective for JK Enterprises' development organization. With RUP, implementation can start earlier to enable early stakeholder feedback, which is necessary to help ensure that all requirements are met. Testing would also start earlier, to enable more defects to be discovered earlier in the process so they're less costly to correct. The project plans will also be refined throughout the project based on a continuous reevaluation of risks and priorities. And with RUP, the development manager and his team will be able to implement an effective development system that is customized to cater to the organization's specific needs.

Given the changing skill demographics of his developers, the development manager also knows that he needs to bring new developers into the System z development process. He uses his offshore distributed developers to develop some of the new CICS applications. He cannot afford the time to make them experts on System z, so he needs a set of application-development and testing tools that they can use intuitively from Rational Application Developer, their existing workstation-based environment.

Although running offshore development teams can be cost effective – something that would please his CFO – the process also requires significantly greater controls. As a result, the development manager requires effective tooling to help maintain the development process, without which he would not be able to meet his governance requirements. Here again, RUP helps by providing his teams with access to the customized development-process definitions, directly from the Web, so they always know what they should be doing during the different stages of the project.

Processes

IBM provides offerings that can help development teams apply iterative development and other modern methods to their SOA projects throughout the life cycle.

Rational Method Composer

IBM Rational Method Composer is a flexible, configurable process platform containing IBM Rational Unified Process and a broad set of methods for the enterprise that go beyond just software development. It is built on top of the open source Eclipse Process Framework (EPF) and contains tooling, process libraries and a unified method architecture for use throughout IT life-cycle management. A Rational Method Composer plug-in for RUP for System z is available, which can generate the RUP for System z Web site. The site describes the development process from beginning to end at different levels of detail.

Rational Unified Process for System z

RUP for System z[2] is a special instance of RUP that provides practitioners with specific software-development guidance and a succinct, end-to-end process specifically geared to the System z environment. RUP for System z includes a large set of work-product examples – mostly produced using IBM tools – taken from an application created in CICS COBOL and exposed as Web services using both the CICS Web Services Assistant and the Enterprise Generation Language (EGL).

RUP for System z addresses development of applications with character-based interfaces. It addresses system evolution that involves architectural changes, including turning an existing capability into a Web service, for instance. And it addresses significant impact on existing business processes. However, pure maintenance activities are outside the scope of RUP for System z. For more information about using RUP for maintenance projects, refer to the IBM Rational Unified Process for Maintenance Projects.[3]

RUP for System z is designed for the entire System z application-development community, from project managers, architects and designers to programmers and testers. It covers the full, end-to-end development life cycle for the System z environment.

RUP for System z is iterative and risk-driven. It focuses on developing the actual software system incrementally in each iteration, as opposed to producing project-related documentation. This facilitates feedback and midcourse corrections, and it supports collaboration among the entire development team. This approach concentrates on using existing assets and on stabilizing the architecture and any high-risk elements early in the project to avoid late discovery of problems that could potentially lead to project failure. It also emphasizes quality by recommending early and frequent evaluation of the running system. Users, customers and other stakeholders are involved in the evaluation process to make sure that the project outcome will be consumable and will meet stakeholders' expectations. In other words, RUP for System z embraces most of the values and practices associated with the agile-development community.

For more information about RUP for System z, see the IBM Redbooks® publication *The Rational Unified Process for System z*.[4] Short extracts from this Redbooks publication, describing the key objectives of the four RUP life-cycle phases, will be used to introduce the following scenarios.

Rational RequisitePro

IBM Rational RequisitePro® is an easy-to-use requirements-management tool that lets teams author and share their requirements using familiar document-based methods while leveraging database-enabled capabilities, such as requirements traceability and impact analysis. The result is better communication and management of requirements with the increased likelihood of completing projects on time, within budget and above expectations. Successful projects start with requirements management – the more effective the execution, the greater the resulting quality and customer satisfaction.

System z development tools

When developing complex, composite and SOA applications on System z, it helps to have a common set of tools that support all of the different languages from COBOL and PL/I to C and C++ and Java™, and target environments from CICS to DB2 and IMS to IBM WebSphere Application Server. IBM System z tools include development tools and problem-determination tools to support the development and testing of all System z applications, and the CICS tools family, which can help to support application transformation and SOA implementation that is based on CICS.

Application-portfolio management

Application-portfolio management is a set of technologies that read source-code artifacts from across the enterprise, record relationships between them in a knowledge base and augment the knowledge base with business information to development management intelligence about applications. Application-portfolio management is used to predict, measure, control and manage the effects of software change.

WebSphere Studio Asset Analyzer for Multiplatforms

IBM WebSphere Studio Asset Analyzer for Multiplatforms provides in-depth insight into dependencies within and among enterprise-application components, including composite applications that span mainframe and distributed components. The product assists IT personnel with the maintenance, extension, reuse and transformation of existing mainframe applications through rapid application understanding and impact analysis. Information provided by WebSphere Studio Asset Analyzer for Multiplatforms can help to modernize existing enterprise assets. For IBM System z and distributed environments, WebSphere Studio Asset Analyzer for Multiplatforms provides information about finding and reusing application code and the components that connect that code. For Java 2 Platform, Enterprise Edition (J2EE) systems, WebSphere Studio Asset Analyzer for Multiplatforms can help analysts and developers understand the code that is operating in WebSphere runtime environments. It can grow with your business to become an integral repository for insight about critical application assets.

Rational Asset Manager

IBM Rational Asset Manager is a collaborative, software-development, asset-management solution that enables organizations to identify, manage and govern the design, development and consumption of software assets, including services, as part of an SOA initiative. The software helps IT organizations deliver innovative IT solutions while controlling costs, reducing application backlogs and improving business flexibility and responsiveness by facilitating reuse of software assets.

WebSphere Service Registry and Repository

IBM WebSphere Service Registry and Repository delivers a robust system for storing, accessing and managing service metadata used in the selection, invocation, management, governance and reuse of services in a successful SOA. This gives development organizations the visibility and control over existing assets for better adaptability and reuse. It helps enforce standards, policies and best practices that help IT groups to get the most business value from their SOA.

CICS Interdependency Analyzer

IBM CICS Interdependency Analyzer for z/OS, Version 2.1 is a runtime tool that automates detection of runtime relationships within a CICS system, records this data in a DB2 database, and provides a easy-to-use GUI (the Explorer) to help analyze the collected information, build a relationship road map and use this data in daily operations. The Explorer provides an easy way to query data stored in DB2, manage queries and navigate through the resource relationships (see Figure 2). This enables developers and other users to perform detailed analysis of resource relationships (for example, what transactions run in which regions, what affinities were found for a program) as well as to compare resources.

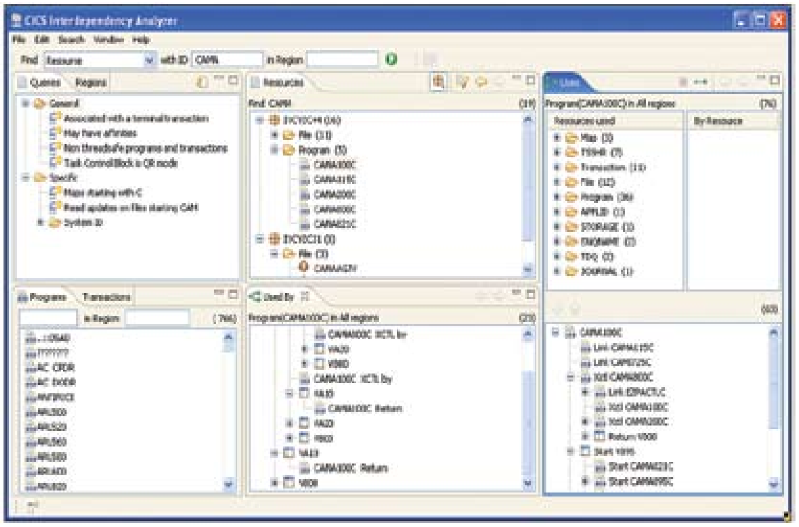

CICS Interdependency Analyzer
File Edit Search Window Help
Find Resource with ID CAMN in Region
Queries Regions
General
Associated with a terminal transaction
May have affinities
Non threadsafe programs and transactions
Task Control Block is QR mode
Specific
Maps starting with C
Read updates on files starting CAM
System ID
Resources
Find: CAMN
File
Program
Uses
Resources used
By Resource
Map
Transaction
File
Program
STORAGE
TDQ
JOURNAL
Programs Transactions
in Region
Used By
in All regions

Figure 2. CICS Interdependency Analyzer Explorer

Application development

Today's applications span multiple platforms and languages, and today's developers require tooling to help them deliver these complex applications as quickly and efficiently as possible.

Rational Developer for System z

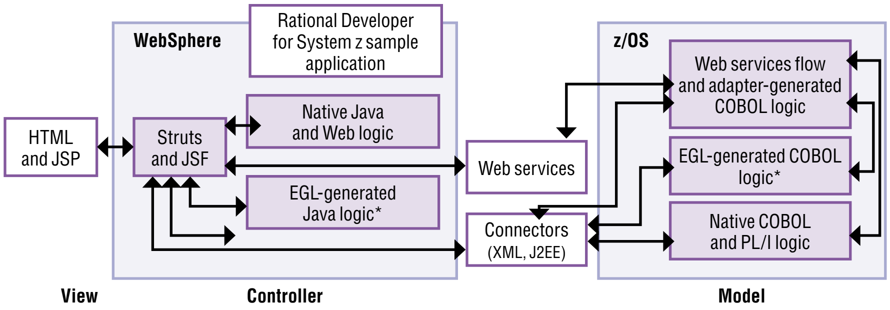
IBM Rational Developer for System z is a workstation-based, integrated development environment (IDE) that helps developers create dynamic Web applications, including support for J2EE, XML and Web-services technologies. It lets developers rapidly create well-structured enterprise applications that integrate WebSphere software and traditional transactional environments, including IBM CICS Transaction Server, IMS and batch systems.

Best practices

Best practices are a powerful supplement to solid processes and powerful tools, helping development teams to simplify many design, development and testing decisions. An effective and common best practice for development is the use of Model-View-Controller (MVC), an application paradigm for SOA. In this paradigm, basic applications are segmented first into three types of processing: visual processing or *view*; control processing, which guides application and business flow; and model processing, which contains the business processing and data. Often today, multiple applications are involved in MVC paradigms. For example, IBM WebSphere® Application Server might provide the visual processing of an application while CICS is often focused on the controller and model components making up SOA.

SOA is based on the concept of reusable services. Some examples of areas in which reusable services can fit into an application include user interfaces, validation, control, microflow, state management, error handling, marshalling and demarshalling and, most importantly, business services. In modern applications, there is a requirement for high-quality processing of these types of logic. Customers have based and continue to base their businesses on these processes.

Figure 1 shows how IBM Rational Developer for System z provides the ability to create and manage all the components in an MVC-structured application.


WebSphere
Rational Developer for System z sample application
HTML and JSP
Struts and JSF
Native Java and Web logic
EGL-generated Java logic*
Web services
Connectors (XML, J2EE)
z/OS
Web services flow and adapter-generated COBOL logic
EGL-generated COBOL logic*
Native COBOL and PL/I logic
View
Controller
Model

*EGL-generated logic requires IBM Rational Business Developer Extension

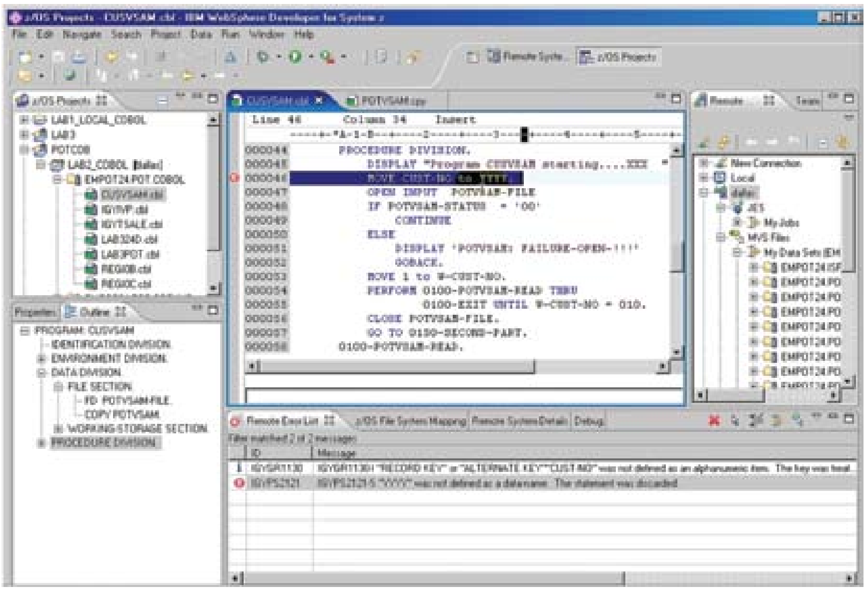
Figure 1. Rational Developer for System z supports the MVC development paradigm.

It promotes the reuse and transformation of existing applications to help reduce costs and shorten the development cycle, and facilitates the need for heterogeneous skill sets in IT organizations by supporting Java, COBOL, PL/I and EGL development. (Note: EGL is provided by IBM Rational Business Developer Extension.)

Rational Developer for System z includes significant capabilities that help make traditional mainframe development, Web development and integrated SOA-based composite development faster and more efficient. It accelerates the development of dynamic Web applications written in Java, J2EE, COBOL and PL/I, and it supports modern user interfaces, full Web-application processing and Web services, and J2EE Connector architecture (JCA) connectivity to integrate these application styles and processes together. Rational Developer for System z supports deployment to multiple runtime environments, including CICS, WebSphere Application Server, IMS, batch and DB2 through stored procedures.

Rational Developer for System z helps application developers apply their existing skills to write new applications, and it promotes the reuse and transformation of existing applications. The productivity of developers who use Rational Developer for System z to create, maintain, debug and deploy applications to the IBM z/OS® platform is further enhanced by integration with other System z tools, such as IBM Debug Tool Utilities and Advanced Functions, IBM Fault Analyzer and IBM File Manager.

Rational Developer for System z enables integration into existing enterprise-application development procedures, such as code management and other existing management procedures. Interfaces to multiple source-code management tools are available as well as interfaces to customize menus, the editor and other aspects of the development environment (see Figure 3).



```
 Line 46      Column 34     Insert
000044       PROCEDURE DIVISION.
000045           DISPLAY "Program CUSVSAM starting....XXX "
000046           MOVE CUST-NO to YYYY.
000047           OPEN INPUT  POTVSAM-FILE
000048           IF POTVSAM-STATUS  = '00'
000049              CONTINUE
000050           ELSE
000051              DISPLAY 'POTVSAM: FAILURE-OPEN-!!!'
000052              GOBACK.
000053           MOVE 1 to W-CUST-NO.
000054           PERFORM 0100-POTVSAM-READ THRU
000055                   0100-EXIT UNTIL W-CUST-NO = 010.
000056           CLOSE POTVSAM-FILE.
000057           GO TO 0150-SECOND-PART.
000058       0100-POTVSAM-READ.
```

Filter matched 2 of 2 messages

ID	Message
IGYGR1130	IGYGR1130-I "RECORD KEY" or "ALTERNATE KEY" "CUST-NO" was not defined as an alphanumeric item. The key was treat
IGYPS2121	IGYPS2121-S "YYYY" was not defined as a data-name. The statement was discarded.

Figure 3. Rational Developer for System z provides both mainframe-based and Web-based tooling for developers.

Rational Business Developer Extension

IBM Rational Business Developer Extension enables IT organizations to build innovative IT solutions while controlling costs, reducing application backlogs, and improving flexibility and responsiveness to the business. Using IBM Rational Business Developer Extension, IT organizations can achieve increased levels of application-development productivity. It also enables business-savvy developers to take full advantage of emerging computing technologies to fulfill new and changing business requirements with minimum retraining costs and time. Rational Business Developer Extension allows organizations to easily reuse and extend valuable existing IT assets. It is an Eclipse plug-in and works with Rational Developer for System z.

IBM Rational Business Developer Extension is the development environment for EGL. The procedural-style programming language is familiar to business-oriented programmers, and is easy to learn and use. It can help eliminate the need to know about underlying middleware and middleware complexities. It supports a variety of operating systems, databases, transaction managers and hardware technologies, and its built-in higher-level abstraction constructs reduce manual coding, and increase productivity and quality.

Application transformation

Application transformation can be defined as identifying and transforming core systems, such as mainframe applications, into modern and flexible applications and services.

Asset Transformation Workbench

IBM Asset Transformation Workbench can help drive down the cost and accelerate the modernization and maintenance of business-critical enterprise applications running on your System z platforms. It helps accelerate time to market, reduce risks and reduce costs associated with modernizing existing applications by providing tooling that helps assess an application's suitability for reuse in an SOA.

IBM Asset Transformation Workbench includes an Enterprise Application Knowledgebase, which provides consistently current, centralized insight into your application portfolio, plus integration with IBM WebSphere Studio Asset Analyzer. Detailed reports, metrics, documentation and visualization of enterprise applications are readily accessible to team members. Asset Transformation Workbench also includes an application analyzer and profiler, and optional business-rule and application-architect extensions. These capabilities can help you identify business rules, as well as begin the process of extracting the rules out of existing applications for use as a Web service.

Host Access Transformation Services

IBM WebSphere Host Access Transformation Services (HATS) delivers tools to extend existing terminal applications quickly and easily to business partners, customers and employees. In addition to transforming the user experience of existing applications with a GUI, HATS can be used to transform and expand the connectivity of existing terminal applications by extending application tasks as Web services. Web services generated with HATS are deployed to WebSphere Application Server.

Also with HATS, CICS terminal-oriented application-screen sequences that are recorded with Service Flow Modeler (a feature of Rational Developer for System z) can be configured and deployed as standard Web services to WebSphere Application Server. HATS can run Web services created with any supported level of CICS Transaction Server. This approach can benefit CICS Transaction Server clients who have an immediate need to integrate CICS 3270 transactions with an SOA using Web services, but who are unable to migrate to CICS Transaction Server, Version 3.1 or have limited availability of CICS application-specialist, solution-development skills. With this approach, the solution assets recorded with Service Flow Modeler can be redeployed directly onto CICS Transaction Server, Version 3.2, enabling configuration and deployment of those assets as Web services without the need to migrate.

Software-configuration management

Software-configuration management is a series of activities that are designed to control change, specifically changes to software assets.

Rational ClearQuest

IBM Rational ClearQuest® offers comprehensive software-change management. It provides defect tracking, process automation, reporting and life-cycle traceability for better visibility and control of the software-development life cycle. Real-time reporting and process enforcement, improved project visibility and control, automated workflows and e-mail notifications can enhance team communication and coordination. Rational ClearQuest also provides test management that unifies development and testing activities, from planning through results. With Rational ClearQuest, you can address the issues of software quality, access control, electronic signatures, repeatable processes and audit trails. It is designed to help simplify compliance management and provides a Web interface that allows easy access from virtually anywhere. Rational ClearQuest facilitates rapid response to change through its integration with tools for requirements, development, building, testing, deployment and portfolio management.

Rational ClearCase

IBM Rational ClearCase® provides sophisticated version control, automated workspace management, and support for parallel development, baseline management, and build-and-release management. It is designed to help improve productivity and operational efficiency with heterogeneous, cross-platform support for distributed, mainframe (IBM z/OS) and midrange (IBM i5/OS®) development. Rational ClearCase enables platform flexibility and enterprise-wide application development. It is accessible through local, remote and Web interfaces, and leading IDEs, including IBM Rational Developer for System z and the Eclipse framework. Its deep integration with requirements, development, build, test and deployment tools provides a complete end-to-end solution to meet current and future needs.

SCLM Advanced Edition

IBM Software Configuration and Library Manager (SCLM) manages changes to z/OS application components, performs auditing and versioning, and controls the movement of your application components. It manages the build processes for generating executable files, organizes components into change sets and performs automatic impact analysis. IBM SCLM Advanced Edition helps z/OS developers to easily approve packages, implement augmented security controls, run three-way merges, simplify the management of SCLM-based projects, and access SCLM services from an Eclipse based development environment, like Rational Developer for System z.

Application-software quality

Managing software quality across the application life cycle requires rigorous testing and enables development groups to identify problems earlier in the life cycle, when making changes is less expensive. IBM application-software quality tools do much more than support traditional applications. They include capabilities that enable you to support SOA-application development, and provide developers and testers with a robust suite of tools that can help improve application delivery throughout the application life cycle. You can use these tools to help increase productivity and IT effectiveness across source-code debugging, application abnormal-end (abend) analysis, and application-performance analysis.

Rational Performance Tester for z/OS

IBM Rational Performance Tester for z/OS enables the use of System z hardware as a load-generation platform. The analysis processes help ensure that applications accommodate loads. It provides a multiuser load- and performance-testing solution for Web-based applications with a browser-like display of each Web page within a test. It provides immediate recognition of performance problems through real-time reporting. Rational Performance Tester can simplify the processes of test creation, load generation and analysis. It permits the insertion of Java code for flexible customization and modeling for diverse user populations. Rational Performance Tester provides Microsoft® Windows® and Linux® technology-based user interfaces that support z/OS platform-hosted test execution with low memory and processor footprints. As a result, even large multiuser tests require limited hardware resources.

Rational Functional Tester

IBM Rational Functional Tester is a tool that provides testers with automated testing capabilities for functional testing, regression testing, GUI testing and data-driven testing. Novice testers are supported with automated capabilities for activities such as data-driven testing, while advanced users have a choice of scripting language and industrial-strength editor. You can choose Java in Eclipse or Microsoft Visual Basic .NET for test authoring and customization. IBM ScriptAssure™ technology and pattern-matching capabilities improve test-script resiliency for frequent changes to an application’s user interface. Version control enables parallel development of test scripts and concurrent usage by geographically distributed teams. Testing of 3270 applications is supported using the IBM Rational Functional Tester Extension for Terminal-based Applications.

Application Performance Analyzer

IBM Application Performance Analyzer is an application-performance monitor for traditional enterprise applications. It allows users to drill down to the application-source level to pinpoint performance bottlenecks. It supports CICS, IMS, WebSphere, DB2 and other z/OS platform-based operating environments. Application Performance Analyzer provides drill-down support for CICS Transaction Server source, and it provides single-user support, allowing simplified access to multiple CICS regions through multiregion operation.

Fault Analyzer

IBM Fault Analyzer for z/OS, Version 8.1 is a robust problem-determination tool that helps you discover why applications fail. It helps you repair these failures quickly by gathering information about an application and its environment at the time of failure. You can also develop and test new and existing applications more productively, helping to reduce costs along the way.

Fault Analyzer works in CICS and most other System z environments and has two modes of operation:

- *Runtime analysis. When a CICS transaction abnormally terminates as an abend, Fault Analyzer produces a report that describes the failure, pinpoints the line of code that caused the abend, formats the variables in use at the time, displays the terminal screen involved, and extracts the message and code details from the appropriate publications. This allows the developer to identify more quickly the root cause. The abend details are also saved in a direct access storage device (DASD) file for subsequent reanalysis or online viewing.*
- *System-dump analysis. If a CICS region creates a dump, Fault Analyzer can read in that dump and analyze it at a system level, showing domain information such as kernel, dispatcher, program-manager and transaction-manager information. The dump information is displayed in a series of panels, which have point-and-shoot hotspots, allowing fast and easy navigation from one functional area to another.*

The latest release of Rational Developer for System z also provides integration with Fault Analyzer, allowing developers to do their abend analysis from their workstation interface.

Debug Tool Utilities and Advanced Functions

IBM Debug Tool Utilities and Advanced Functions is a source-level debugger that works in both CICS and other environments. It allows application programmers to step through an IBM Language Environment® application written in COBOL, C, C++, PL/I or assembler that is enabled for use in a Language Environment context. As programmers step through program logic, they can display program variables and application data to understand how the application is behaving. Debug Tool Utilities and Advanced Functions also has a comprehensive range of break points, which allows the user to run the application at near full speed until the occurrence of a specific event, such as entry to a specific piece of code. The developer can use the 3270 interface or a GUI (available with a separately purchased license of IBM WebSphere Developer Debugger for System z or Rational Developer for System z) to help deliver code that is more-thoroughly tested and has fewer errors.

Workload Simulator for z/OS and OS/390

IBM Workload Simulator for z/OS and OS/390® enables you to conduct planning tests for stress, performance, regression, function and capacity, while eliminating the need for large amounts of terminal hardware and operator time. It provides support for Systems Network Architecture (SNA), Common Programming Interface for Communications (CPI-C) LU 6.2 and enhanced Transmission Control Protocol/Internet Protocol (TCP/IP). It also supports Telnet 3270, 3270E and 5250 clients; Telnet line-mode network virtual terminal (NVT) clients; simple TCP and User Datagram Protocol (UDP) clients; File Transfer Protocol (FTP) clients; and multiple client applications that run on top of TCP/IP.

Additional tools for System z developers

IBM offerings are available for file management, test-data generation and CICS resource definitions in a z/OS environment.

File Manager for z/OS

IBM File Manager for z/OS helps the developer to create, edit, print and format or reformat data files in the most-popular z/OS file formats. It enables programmers to manipulate data using COBOL and PL/I record layouts in batch and an online facility. File Manager includes support for Virtual Storage Access Method (VSAM), CICS, DB2 and IMS data. It can also be integrated within the Rational Developer for System z development tool.

File Export for z/OS

Populating test environments with data or migrating application data to different environments can be complicated. You know and understand the data relationships, but getting different sources together is time consuming. IBM File Export for z/OS can make this process quick and easy.

IBM File Export for z/OS is a flexible solution for exporting and importing related sets of enterprise data. It copies all data or just a slice of data in a related set of objects and delivers supports for DB2, IMS and MVS™ data (including VSAM) in a single product. It supports application-defined relationships and provides for one-to-one, one-to-many and many-to-one data creation. The integrated repository stores common persistent information about your data, including data-scrambling rules that let you protect sensitive data, field by field. With this powerful tool, you can generate test data when you need it.

CICS Configuration Manager

Accurate CICS resource definitions are essential to maintain the high availability expected of the CICS environment. With tens, or even hundreds, of CICS regions across a typical enterprise, maintaining these definitions can be challenging. IBM CICS Configuration Manager for z/OS simplifies CICS resource-definition administration and maintenance, while at the same time offering comprehensive reporting and optional change-management control facilities. It enables the control of all CICS definitions across the enterprise through a single interface and can manipulate definitions seamlessly across CICS system definition (CSD) files and IBM CICSPlex® System Manager data repositories. It enables integration with existing system-management processes, and helps improve auditability and ease CICS version upgrades.

Using IBM System z tools to develop SOA applications—a scenario

Let’s return to JK Enterprises for another scenario. JK Enterprises is a midsized company with eight data centers and about 200 IT staff. The staff members rely on CICS to host many of their mission-critical business applications with several hundred CICS regions, of which one hundred are in production. They have a mixture of build-and-buy applications that run in WebSphere Application Server, SAP, Siebel, CICS, and numerous batch applications.

A strategic business goal of JK Enterprises is to generate additional revenue and market share but, as always, with cost constraints. The company’s CIO has recently come back from a meeting with other CIOs, where he heard many success stories about SOA and its benefits. To meet the company’s strategic business goal, JK Enterprises will give customers and business partners access to its order processing and inventory applications. Based on what the CIO learned at his meeting, he firmly believes that much of the current application inventory can be reused in this new process. Now the IT group has to make this happen.

At a high level, here are the seven steps that the SOA project team has identified:

1) *Decompose the business need or goal into required high-level functions and services. To do this, the team members know they will need to create a Web site, provide security for logging on, inquiry access to the mainframe-based inventory system, and access to the individual customer information, which is also mainframe based. They will also need to create an ordering form that can interface to the mainframe-based ordering system.*
2) *Analyze the existing applications. From their analysis of the business need, the team members know that the mainframe-based inventory, customer records and ordering systems will be involved in this new application.*

3) *Identify the actual programs and data items that are part of the existing systems that could also be used in this new application. Most of the programs that the team wants to use in the new application include 3270 terminal maps and error handling, as well as the data-access and business code. Because these programs have been written and maintained over a number of years, there is no available, up-to-date documentation about them.*

4) Enable the existing code as services. This could be achieved in either of these two ways: They decide to use a combination of these methods.

- *Because they are a CICS shop, the team could package the existing transaction in a wrapper and make it into a Web service.*
- *They could extract the code from the existing application and make it a stand-alone program that can then be used by both the existing 3270 terminal-based application and the new Web-based application.*

5) *Create some new services. In particular, for security and to provide several of the new interfaces, the team will need to write some code.*

6) *Test the new and modified applications. Load and performance testing are required in addition to functional testing to help ensure that the system can support its new external users.*

7) *Deliver the new service-oriented ordering and inventory system to customers and business partners.*

Step 1: Decompose the business need or goal into required high-level functions and services. To do this, the team members know they will need to create a Web site, provide security for logging on, inquiry access to the mainframe-based inventory system, and access to the individual customer information, which is also mainframe based. They will also need to create an ordering form that can interface to the mainframe-based ordering system.

We will follow JK Enterprises through its use of key RUP for System z processes to show how some of the IBM tools described in the previous section can help the team to plan, develop and test changes to its CICS and other System z applications. Our scenario will follow the same seven steps mapped to RUP for the System z life-cycle phases of inception, elaboration, construction and transition.

Inception phase, Step 1—requirements gathering

The overriding goal of the inception phase is to achieve concurrence among all stakeholders on the life-cycle objectives for the project. The inception phase is of significance primarily for new development efforts, in which there are significant business and requirement risks, which must be addressed before the project can proceed. For projects focused on enhancements to an existing system, the inception phase is shorter but is still focused on ensuring that the project is both worth doing and possible to do.[4]

Manage the requirement process with Rational RequisitePro

JK Enterprises chooses to use IBM Rational RequisitePro to enable its global teams to clearly understand and effectively manage changing requirements in their natural context. With IBM Rational RequisitePro, JK Enterprises can reduce rework by improving the communication and interpretation of its global requirements. Rational RequisitePro provides the mechanism to document the company's requirements and map them to subsequent development tasks, giving the project-management team a high degree of confidence that the business objects of the project are met.

Manage the development process with Rational ClearCase and ClearQuest, and SCLM Advanced Edition

JK Enterprises wants to benefit from being a multisite organization but avoid the lack of control that often comes with distributed development. At the same time, the company does not want its teams to be so restricted that they look for ways to bypass the system. JK Enterprises chooses Rational ClearQuest and Rational ClearCase software, because the development and integration models, private workspaces and public integration areas of the software allow developers to work independently while collaborating effectively with the team. In addition, the user-authentication and audit trails can help the team meet its compliance requirements with minimal administrative hassles. And with access virtually anytime, anywhere, Rational ClearCase gives the developers the freedom to work efficiently where and when needed. ClearCase also interfaces to Rational Developer for System z, so the developers do not need to change tools for such tasks as checking out and checking in application code.

Although JK Enterprises decided to primarily use ClearQuest and ClearCase for managing development across its platforms, including z/OS, the company has also identified some z/OS applications that do not need to be managed under a consolidated infrastructure. For those applications, JK decides to use SCLM and take full advantage of the ability to integrate with ClearQuest. This integration allows the development team to follow a more-traditional development while still coordinating their efforts with other development teams.

Step 2: Analyze the existing applications. From their analysis of the business need, the team members know that the mainframe-based inventory, customer records and ordering systems will be involved in this new application.

Step 3: Identify the actual programs and data items that are part of the existing systems that could also be used in this new application. Most of the programs that the team wants to use in the new application include 3270 terminal maps and error handling, as well as the data-access and business code. Because these programs have been written and maintained over a number of years, there is no available, up-to-date documentation about them.

Elaboration phase, Steps 2 and 3—analysis and design

The goal of the elaboration phase is to establish a baseline architecture for the system, providing a stable basis for the bulk of the design and implementation effort in the construction phase. The architecture evolves out of a consideration of the most significant requirements (those that have a great impact on the architecture of the system) and an assessment of risk. The stability of the architecture is evaluated through one or more architectural prototypes.[4]

Relationship discovery as an aid to understanding applications

Making changes to applications without understanding their relationships and then making changes to related objects is a recipe for failure. Without good documentation, extensive manual work would be required to determine whether a program could or should be reused. However, JK Enterprises chooses to use tools such as CICS Interdependency Analyzer, WebSphere Studio Asset Analyzer and Asset Transformation Workbench. These tools help the company analyze its existing applications and identify those assets that could be leveraged as services.

JK Enterprises uses WebSphere Studio Asset Analyzer to create a centralized application-metadata repository that provides a foundation for application insight. This helps them understand their application assets and their dependencies, and gives them up-to-date documentation to speed knowledge transfer. The centralized repository helps them find sets of programs and related assets that embody specific business tasks and also gives project leaders visibility into code quality as well as complexity hotspots.

In order to provide external access to its systems, the analysis done by JK Enterprises indicates that the development team needs to change the layouts of some VSAM files, DB2 tables and IBM WebSphere MQ messages to reflect new access paths, order statuses and other functions. CICS Interdependency Analyzer provides insight into dynamic, runtime, CICS resource relationships. WebSphere Studio Asset Analyzer provides a deep understanding of such static resources as JCL, application source code and include files. As a result, the company’s analysts, developers and testers gain the ability to understand the underlying relationships for all of their System z application assets.

To size projects better, the development team determines the effect of a change before making the change, and estimates resource requirements based on current application metrics. All of this knowledge helps the team to plan for the application-transformation project, as well as for future maintenance efforts. The team will be also able to identify business rules so that that it can service-enable other application assets more quickly.

Locate SOA artifacts with Rational Asset Manager

Rational Asset Manager helps the JK Enterprises development team to create, discover and trace service assets throughout the SOA life cycle and to increase developer productivity while promoting the reuse of assets related to software development. Rational Asset Manager also helps them monitor asset integrity and utilization through a defined, enforceable and auditable process that includes fine-grained permissions and review-approval flows.

Rational Asset Manager also integrates with IBM WebSphere Service Registry and Repository software and other IBM software, including Rational ClearCase, Rational ClearQuest for change management and IBM Rational Architecture Management software for asset development.

JK Enterprises had not started to use WebSphere Service Registry and Repository at the start of this project, but recognizes the value it could bring not just to their development process but also to help support operational use of Web services and to support their future Web service governance requirements.[5] So, they plan to implement it after their first development is complete. For more details about Rational Asset Manager and WebSphere Service Registry and Repository integration, see the referenced white paper.

Step 4: Enable the existing code as services. This could be achieved in either of these two ways:

- *Because they are a CICS shop, the team could package the existing transaction in a wrapper and make it into a Web service.*
- *They could extract the code from the existing application and make it a stand-alone program that can then be used by both the existing 3270 terminal-based application and the new Web-based application.*

They decide to use a combination of these methods.

Step 5: Create some new services. In particular, for security and to provide several of the new interfaces, the team will need to write some code.

Construction phase, Steps 4 and 5—development and unit testing

The goal of the construction phase is to clarify the remaining requirements and complete the development of the system change to: using the baseline architecture. The construction phase is, in a sense, a manufacturing process, in which emphasis is placed on managing resources and controlling operations to optimize costs, schedules and quality. In this sense, the management mindset undergoes a transition from the development of intellectual property during inception and elaboration, to the development of deployable products during construction and transition.[5]

Development of a Web-service wrapper with HATS

JK Enterprises needs to evaluate each of the programs that it identified as required for its new Web-based system, and determine how best to develop the system. Because CICS Interdependency Analyzer and WebSphere Studio Asset Analyzer for Multiplatforms identified all the components (during the elaboration phase), Asset Transformation Workbench can the help the JK team with in-depth analysis of its existing applications. Asset Transformation Workbench uses the WebSphere Studio Asset Analyzer bridge (which is part of Asset Transformation Workbench) to download the assets previously identified by WebSphere Studio Asset Analyzer. Asset Transformation Workbench provides in-depth analysis that will help them better understand which assets they can most easily reuse. Because of the complexity of the inventory system, the team decides to use HATS as one of the quickest ways to enable an existing application for Web services.

HATS can help the team transform complex, CICS 3270 terminal-based application interactions so that they can be accessed as HTML Web pages or as Web services. Enabling existing code as services can be accomplished in one of several ways. The customer-data system of JK Enterprises is a CICS application, and – as a CICS Transaction Server, Version 3 userthe team has several options. JK decides to use HATS to create adapters to integrate their existing customer applications into the new Web applications. The adapters link together numerous 3270 interactions from multiple internal systems to create intuitive yet powerful Web applications for their new users. For more information about this approach, see the IBM Redbooks publication *Host Access Transformation Services on z/OS.* [6]

COBOL Web-service development with Rational Developer for System z and Rational Business Developer Extension

JK Enterprises needs a new customer-oriented order application as well as updated back-office processing that takes into account their new user base. The company decided to create a sales portal on the Web with Web-based back-office processing of customer data.

In the case of the order system, the team decides that it makes sense to refactor the application by extracting the code from the existing application, and then make the code a stand-alone component that can be used by both the existing 3270 terminal-based application, as well as the new Web-based application. The team refactors the order system by using Asset Transformation Workbench to extract the applicable source code, and then using Rational Developer for System z tooling to enable it as a Web service. All of this is done using the IBM Enterprise COBOL compiler.

Rational Developer for System z, with the Rational Business Developer Extension plug-in provides highly productive tools to design Web Pages in HTML, and to link Web clients to framework processing in JavaServer Faces (JSF), EGL and Java. It helps the development of Web services through analysis of COBOL and PL/I processing, and it can integrate CICS business flows (microflows), tying programs and screens together as processes. It enables the rapid development of CICS COBOL, PL/I, EGL, C and C++ by linking workstation productivity with mainframe asset and job management. Powerful distributed debugging gives the developer a full end-to-end view of the applications, wherever the applications might run.

Rational Developer for System z gives the JK Enterprises application developers a modern development environment that supports the new technologies that make up SOA. It provides them with the ability to develop new Web and rich-client user interfaces written in JSF, J2EE, EGL and JavaScript™, and then link the interfaces through Web services to high-performance, business processing running on CICS Transaction Server, IMS, DB2 and other System z environments.

For the new back-office processing, JK Enterprises plans to refactor some of its existing CICS application programs to enable them to be accessed directly as Web services, by using the CICS Service Flow Feature together with the Service Flow Modeler of Rational Developer for System z to componentize CICS microflows.

Test data creation with File Manager

For JK Enterprises, changes to applications also involve changes to the layout of records and databases. In such cases, it's usually necessary to populate the test files and databases with data that matches the new formats. The edit, copy, global-change and export capabilities of IBM File Manager enable the company's development and function-test teams to generate test data sets quickly.

IBM File Manager also helps the developers to introduce complex conditions, logical inconsistencies and other errors into the test data to validate that the applications will function correctly, should those circumstances arise in production. They're also able to create valid records in the new formats before all of the modified applications are completed. This enables testing of some of the simpler reporting programs to be performed out of sequence, which works well with the team's iterative development process and results in a shorter overall test phase.

The database administrator is able to use the interactive SQL SELECT prototyping utility in the File Manager DB2 component to help her build valid SELECT statements. She is able to test, analyze and run SQL statements interactively and then browse or edit the formatted result table and save the statements for use by the developers in their application code.

JK Enterprises discovers that the power and flexibility of IBM File Manager and its integration with Rational Developer for System z is a great complement to the iterative development cycle advocated by RUP. The ability to create test data easily, quickly and well before the data-conversion programs have been written means the team is able to deliver its beta and alpha releases ahead of schedule.

Unit testing

As the team goes through its iterative development process, the developers often use the debugger in Rational Developer for System z to test code changes. They're able to do this because of the integration of the tool with the Debug Tool Utilities and Advanced Function that is installed on the mainframe, which allows them to test any of their mainframe-code changes. Rational Developer for System z also provides them with the ability to test Web-service code using the same debugging interface.

To accelerate their development process, the developers at JK Enterprises use Rational Functional Tester to record user interactions for their new Web-based applications, creating a test script which, when run, will automatically reproduce those actions. During recording, the developers insert verification points to extract specified data or properties from the application under test. During playback, they use these verification points to compare recorded information with live information to ensure consistency. For some complex tests, they use the option of adding custom code to the test script to perform custom data manipulation.

The team is able to test changes to the existing 3270 applications using IBM Rational Functional Tester Extension for Terminal-based Applications. This enables the developers to use the same tool and share scripts for both types of applications. Following test execution, the developer can use reports generated by Rational Functional Tester to demonstrate to the JK development manager that his applications had been properly tested.

With Rational Functional Tester Extension for Terminal-based Applications, the company's development teams are able to expose problems more reliably and efficiently in even the most-complex applications. And they're able to increase the opportunity for defect capture and repair before product release, even with parallel development by geographically distributed teams.

Diagnostics and root cause analysis using Fault Analyzer

Problems resulting from erroneous application design or code are hard to find, but IBM Fault Analyzer can help developers diagnose and find the root cause of complex System z problems, whether they occur in a CICS, IMS or batch environment. When an abend has occurred and a dump is about to be written, Fault Analyzer is started through the OS/390 Change Options/Suppress Dump exit list. At this point, Fault Analyzer performs a real-time analysis of the storage within the abend region. The process records the job name, abend code and other such information in the dump catalog, along with the analysis report. If the analysis has found sufficient information for the report (and the RETAINDUMP option is set to AUTO), the pending dump is suppressed. Otherwise the dump is written and the data-set name is included in the catalog record for the dump. After the abend, an appropriate developer (in a test situation) or operator is notified by the job, depending on your installation procedures.

The distributed members of the JK development team, who are relatively new to System z, find Fault Analyzer's integration with Rational Developer for System z especially useful, because it let them drill quickly to the root cause of the problem without having to leave their development environment or use ISPF panels.

Locating failure points with Debug Tool Utilities and Advanced Functions

IBM Debug Tool Utilities and Advanced Functions helps JK developers on a day-to-day basis. They use its source-level debugging through Rational Developer for System z integration, which enables them to track down the causes of application failure, such as those due to data-format inconsistencies. Real-time changes to the application and the data it uses can be made and the job restarted where it left off. This integration of Rational Developer for System z and Debug Tool Utilities allows developers to see such specifics as the failure point within the program and the value and storage of fields. For example, they could see that a field such as CustomerNumber was defined as alphabetic characters where numerics are expected.

Early warning of potential performance problems with Application Performance Analyzer

The JK Enterprises development team understands that most performance issues are introduced early in the life cycle, so they developed a best practice to validate application performance during the early prototyping and development phases. The developers use IBM Application Performance Analyzer, typically starting with a Measurement Profile Report to provide an at-a-glance performance summary. Intuitive chart views give them fast visual recognition of performance hot spots with convenient linking to other reports to access additional detail on each report section. The Measurement Profile Report contains summaries of key z/OS resources, including CPU, DB2 (including most-active plans, DBRM, and active SQL statements), paging, EXCPs, CICS, IMS and WebSphere MQ.

Step 6: Test the new and modified applications. Load and performance testing are required in addition to functional testing to help ensure that the system can support its new external users.

Step 7: Deliver the new service-oriented ordering and inventory system to customers and business partners.

Using Application Performance Analyzer, the JK Enterprises developers can locate and eliminate several design problems that would be harder to fix if not found until the performance test, and which would significantly affect performance if the applications were deployed unchanged in production.

Transition phase, Steps 6 and 7—system testing and deployment

The focus of the transition phase is to ensure that software is available for its users. The transition phase can span several iterations, and includes testing the product in preparation for release and making minor adjustments based on user feedback. At this point in the life cycle, user feedback needs to focus mainly on fine-tuning the product, configuring, installing and usability issues. All the major structural issues need to have been worked out much earlier in the project life cycle.[4]

Knowing which applications to test

Changing the layout of files, tables and messages usually, though not always, requires modifications to programs that access them. But understanding such relationships in a complex and mature application can be difficult, especially when all of the original developers might have left the company, the source code might have been lost, and key components might have been created by third parties or not documented according to company standards.

The development team members at JK Enterprises had recently purchased CICS Interdependency Analyzer for z/OS and had run the tool's Collector for some weeks against their production systems, enabling them to build a comprehensive database of CICS resource relationships. The team members then used the new Eclipse-based Explorer of CICS Interdependency Analyzer to understand quickly and easily what transactions, programs and maps would be affected by the changed file formats, enabling them to test all related applications for quality assurance.

Testing and validating performance objectives

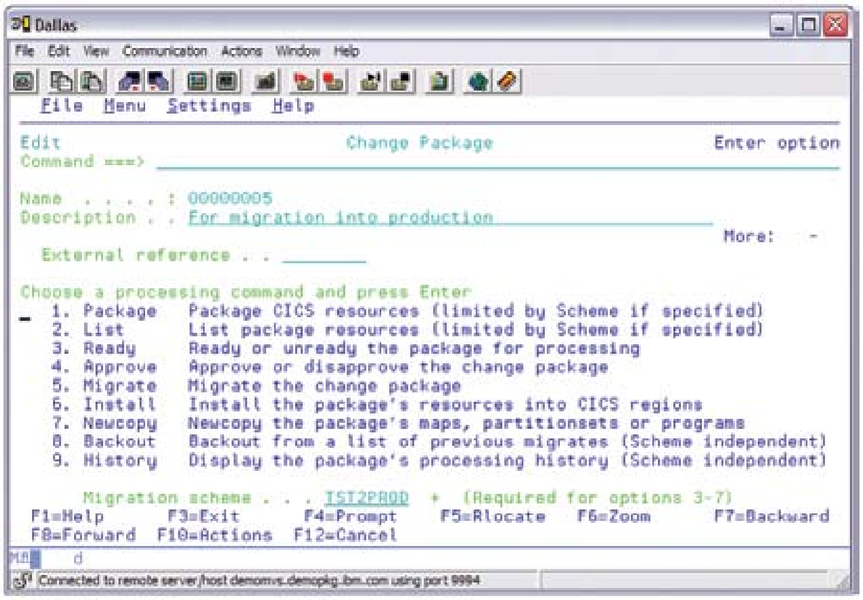
Having successfully completed the system test, the JK Enterprises quality-assurance team plans to run a performance test to validate that the new and modified applications will stand up to the increased load from the external customers and business partners. They recognize that they'll need near-production-scale databases and VSAM files if they are not to be misled by data caching or contention. But they're not prepared to simply copy production files, because the files contain sensitive customer data. They decide to solve this problem by using IBM File Export, which not only creates extracts from VSAM files and DB2 and IMS databases, but also provides powerful data-scrambling capabilities to disguise sensitive information. File Export provides a persistent repository of record layouts, fields and relationships, as well as repeatable scrambling algorithms. This enables referential integrity to be maintained across multiple files and databases.

Because the JK Enterprises team members need to test the new and existing interfaces for their changed applications, they decide to use Rational Performance Tester for the new Web applications and IBM Workload Simulator for the modified 3270 terminal-based applications. Both of these performance-testing products enable them to use available System z capacity to generate test loads.

After running the tests, the team members use CICS Performance Analyzer to validate that the performance meets their predefined objectives. Of special value are the new Transaction Profiling reports, which enable them to compare performance easily before and after the application changes. The reports highlight a performance anomaly in one particular transaction. Detail and focused CICS Performance Analyzer reports then show that some instances of this transaction performed excessive numbers of DB2 requests. By linking the poorly performing instance to a specific time of day and terminal ID, the development team identifies the particular test case that triggered the problem and then locates the root cause.

Changing CICS configuration

JK Enterprises manages and controls not only its application programs, but also the configuration information related to those applications. CICS Configuration Manager automates and manages configuration changes to CICS resource definitions and ensures that CICS configuration changes are made in a controlled and authorized manner, which can reduce the risk of human error. The JK Enterprises release-management team is able to create change packages that contain all of the resource-definition changes for the new application release so that the resource-definition changes can be deployed or backed out in a single operation. (See Figure 4.) By creating separate packages for the customer, order and back-office applications, the team is able to deploy or back them out individually, giving them greater flexibility. The JK Enterprises system programmer also saves time by using CICS Configuration Manager automation to convert program names and names of VSAM data sets to meet local and regional standards automatically as the release moves through the different phases, such as from development to system test, then to performance test and finally to production.



```
  File  Menu  Settings  Help

Edit                        Change Package                     Enter option
Command ===> ______________________________________________________________

Name . . . . : 00000005
Description . . For migration into production
                                                              More:    -
  External reference . . ________

Choose a processing command and press Enter
_  1. Package   Package CICS resources (limited by Scheme if specified)
   2. List      List package resources (limited by Scheme if specified)
   3. Ready     Ready or unready the package for processing
   4. Approve   Approve or disapprove the change package
   5. Migrate   Migrate the change package
   6. Install   Install the package's resources into CICS regions
   7. Newcopy   Newcopy the package's maps, partitionsets or programs
   8. Backout   Backout from a list of previous migrates (Scheme independent)
   9. History   Display the package's processing history (Scheme independent)

      Migration scheme . . . TST2PROD  +  (Required for options 3-7)
 F1=Help      F3=Exit      F4=Prompt    F5=Rlocate   F6=Zoom      F7=Backward
 F8=Forward  F10=Actions  F12=Cancel
MA     d
Connected to remote server/host demomvs.demopkg.ibm.com using port 9994
```

Figure 4. CICS Configuration Manager — change package

Deploying the new release

After rerunning the CICS Interdependency Analyzer Collector in the test regions, the developer architect uses the CICS Interdependency Analyzer Explorer to highlight the differences before and after the changes. The release specialist also uses CICS Interdependency Analyzer to generate the CICSPlex System Manager workload-manager definitions for the new updated application for direct input to CICSPlex System Manager. This not only saves time but also reduces the chance that errors will be made. He also uses CICS Configuration Manager to control the installation of the new and updated resource definitions into the production system with the assurance that, if necessary, CICS Configuration Manager can back out the definition changes easily if problems occur when the new release is deployed.

Tuning the new release

After the updated application has been running in production for a week or two, JK Enterprises uses CICS Performance Analyzer to look for tuning opportunities. CICS Performance Analyzer highlights some CICS transaction clustering that indicated possible affinities that would affect workload-management in the production system. The team then uses CICS Interdependency Analyzer to track down the affinities which, although not causing the *applications* to perform badly, would affect the ability of the *system* to perform optimally. These affinities are reported to development and a plan is put in place to remove them during the next application-update cycle.

Supporting the new release

After the new release has been deployed, it becomes the domain of IT service management. The role of processes (in this case, based on IT Infrastructure Library®) and of the System z tools that support them is described in a companion white paper *IBM service management for CICS with System z tools*.[1]

Conclusion

The challenge facing IT development today is to do more and to do it better but with fewer failures, more control and a smaller (and often less-highly skilled) team. The old saying "If you want different results, you need to do things differently" still applies. The different approach recommended here is to combine a formal, process-oriented application-development system, as defined by Rational Unified Process for System z, with powerful tooling, such as that in the IBM System z tools portfolio. Be smart like JK Enterprises; do more, with less. And do it right.

(Note: In our fictitious example, we describe a scenario in which a company converts to an SOA in a revolutionary manner. In reality, these steps would, could and should be done in an evolutionary manner, in stages over a period of time. Not only that, but each company will have its own path to SOA. IBM can help you identify the proper road map for your move to SOA.)

For more information

To learn more about IBM Rational Unified Process, visit:

ibm.com/software/awdtools/rup

To learn more about any of the development tools described in this paper, visit:

ibm.com/software/info/developer/index.jsp

To learn more about any of the problem determination tools, visit:

ibm.com/software/awdtools/deployment/

To learn more about CICS Tools, visit:

ibm.com/cics/tools/

Many of the products in the CICS Tools family can be downloaded and used for evaluation for up to 60 days, without commitment to purchase. For more information and for trial registration and product download,visit:

ibm.com/software/os/zseries/trials/cicstools/

To learn more about IBM System z software, visit:

ibm.com/software/os/systemz

To learn more about best practices, replay the teleconference *Best practices for managing the performance of composite applications* by visiting:

ibm.com/software/os/systemz/telecon/25may/

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

Produced in the United States of America
10-07
All Rights Reserved

[1] Knutson, John. 2007. "IBM service management for CICS with System z tools." IBM, ftp://ftp.software.ibm.com/software/htp/cics/tools/WSW11330-USEN-00_cicstools_itsm_WP_0619D.pdf (accessed September 7, 2007).

[2] Carroll, Kathy, Mike Edwards, Angelo Fernandes, Enrico Mancin, and Cecile Peraire. The IBM Rational Unified Process for System z: Making development on System z more agile. IBM, http://www.ibm.com/developerworks/rational/library/jun07/peraire/index.html?S_TACT=105AGX15&S_CMP=EDU#notes#notes (accessed September 7, 2007).

[3] IBM developerWorks® Web site. RUP for Maintenance Projects Plug-in V1.0. IBM, http://www.ibm.com/developerworks/rational/downloads/06/plugins/rmc_prj_mnt/ (accessed September 7, 2007).

[4] Carroll, Kathy, Mike Edwards, Angelo Fernandes, Enrico Mancin, and Cecile Peraire. The IBM Rational Unified Process for System z. IBM, http://www.redbooks.ibm.com/redpieces/abstracts/SG247362.html?Open (accessed September 7, 2007).

[5] Larsen, Grant, and Marc-Thomas Schmidt. "Federated metadata management with IBM Rational and WebSphere software." IBM, ftp://ftp.software.ibm.com/software/rational/web/whitepapers/10709147_Rational_RAM_WP_ACC.pdf (accessed September 7, 2007).

[6] DeCarlo, Joe, and Scott G. Peters. Host Access Transformation Services on z/OS. IBM. http://www.redbooks.ibm.com/abstracts/sg246479.html (accessed September 7, 2007).