



Modern Application Architectures for COBOL Developers - An Introduction

WebSphere software



Agenda

- COBOL Today
- Service Oriented Architecture
 - Introduction
 - Challenges for System z Customers
 - Strategies
- SOA and the System z Application Lifecycle



COBOL Today and the future

- COBOL (COmmon Business Oriented Language)
 - The predominant programming language of business applications for over 40 years
 - Specifically designed for business applications
 - Two million programmers write up to 5 Billion lines of COBOL code every year.
- The following factors are some of the reasons that COBOL continues to maintain its reign as the predominant programming language for commercial business applications.
 - Strong presence of COBOL vendors
 - Modern COBOL extensions to existing COBOL applications
 - COBOL's ease of use and ease of comprehension reduces documentation and learning costs.
 - Continues to be popular and its use is growing
 - IBM continues to deliver value in its COBOL compiler products.
 - COBOL is easy to learn and maintain over time, with or without formal training.
 - The mainframe delivers superior operational efficiency due to its centralized design.
 - Offloaded applications would increase the costs of operations
 - Effort of offloading applications off the mainframe is risky and expensive.
 - Migrating COBOL off the mainframe can cost \$25 per line of code (Network World Oct 20, 2003).



What is Service Oriented Architecture (SOA)?

... a service?

A **repeatable business task** – e.g., check customer credit; open new account

... service orientation?

A way of integrating your **business as linked services** and the outcomes that they bring

... service oriented architecture (SOA)?

An IT **architectural style** that supports service orientation

... a composite application?

A set of **related & integrated** services that support a business process built on an SOA

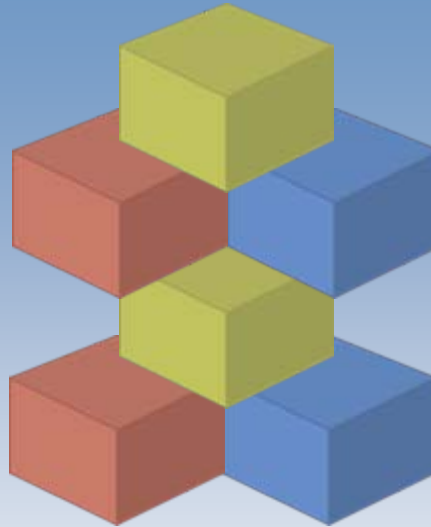


SOA: The focus is on Flexibility and Reuse

Business Perspective

Modern UI's linked with Business Process

- Orchestrated sequence of
 - Activities
- Separated elements
 - Activity sequence
 - Activity hand-off
 - Activity content



IT Perspective

Web User Interfaces and Composite Application

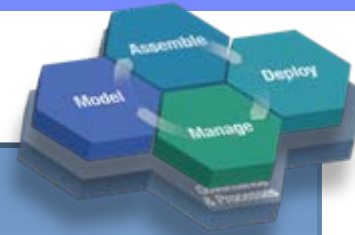
- Orchestrated flows of Services
 - Tooling
- Separated logic
 - Process flow
 - Connectivity
 - Business
- Flexible high QOS Business Functions

Why Service Oriented Architecture? ...

- Enables re-use of existing assets
- Enhances system flexibility through logic isolation
- Supports simplified integration of new assets with existing assets

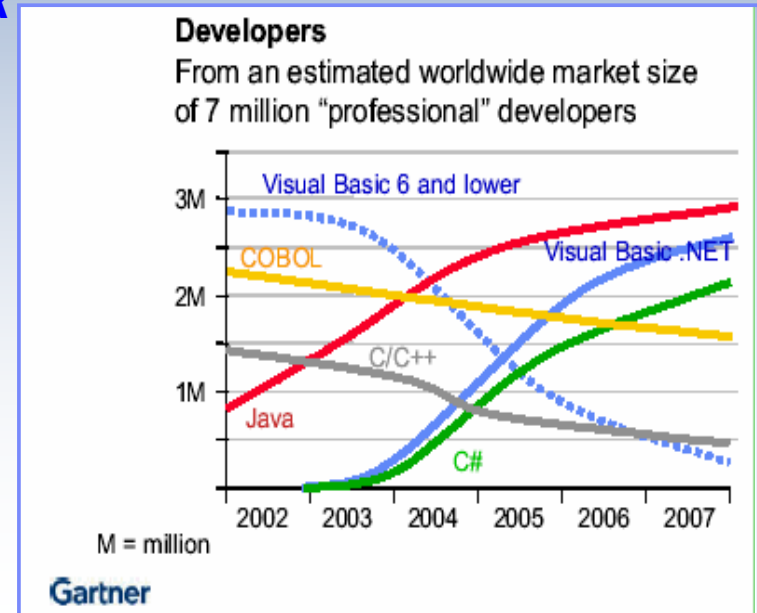


What about “before SOA”?



- Significant business intelligence exists in core systems
 - “200 Billion lines of COBOL code in existence” **eWeek**
 - “5 Billion lines of COBOL code added yearly“ **Bill Ulrich, TSG Inc.**
 - “2 Million COBOL developers” **Gartner**
 - “Majority of customer data still on mainframes“ **Computerworld**
 - “Replacement costs \$20 Trillion” **eWeek**

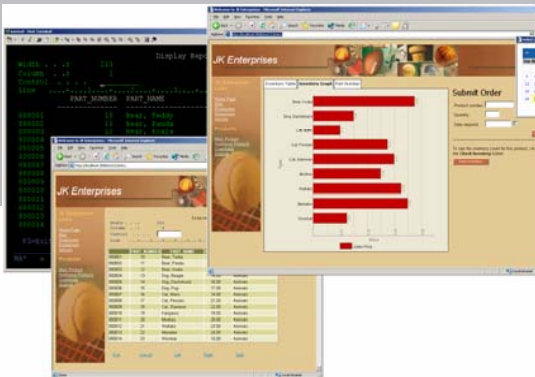
- Rewriting - is it an option.....
 - How long will it take? (lose strategic benefit)
 - Who will do it? (who has the business knowledge?)
 - How much will it cost?
 - Risk?



Three Styles of Application Transformation

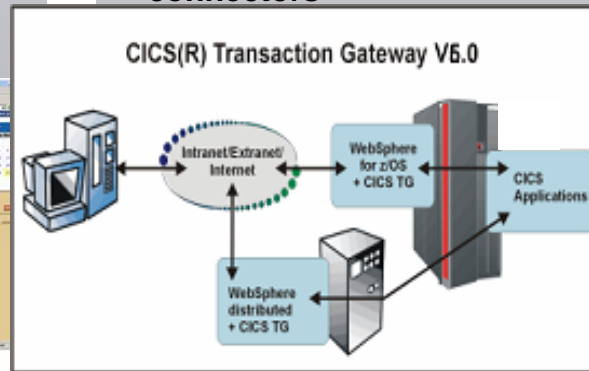
Transform User Experience

Enhance user interface and workflow for quick return on investment



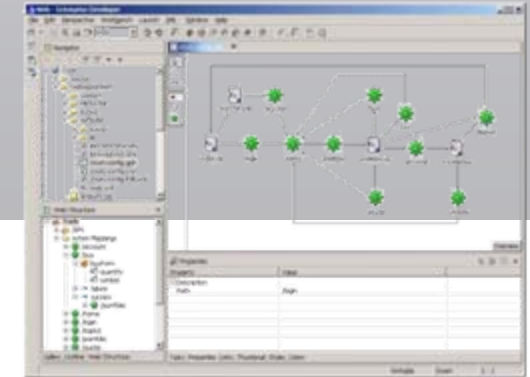
Transform Application Connectivity

Improve business processes and develop customer, partner and supplier relationships using Web services and Java connectors



Transform Application Architecture

Update and extend mission-critical applications as services, leveraging their core value in new ways



Single integrated delivery vehicle across application transformation styles

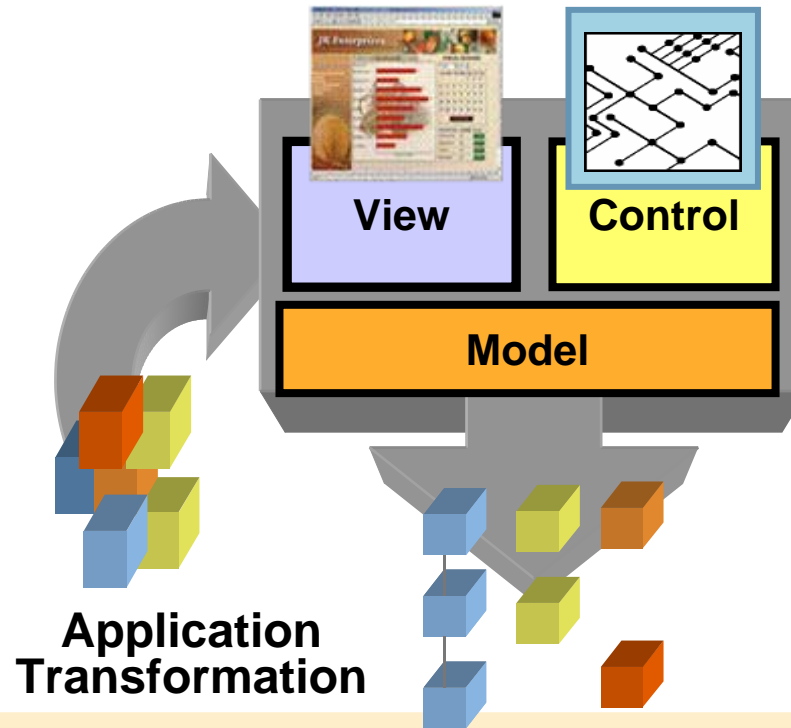


Three styles of Application Transformation

Transform
User Experience

Transform
Application Connectivity

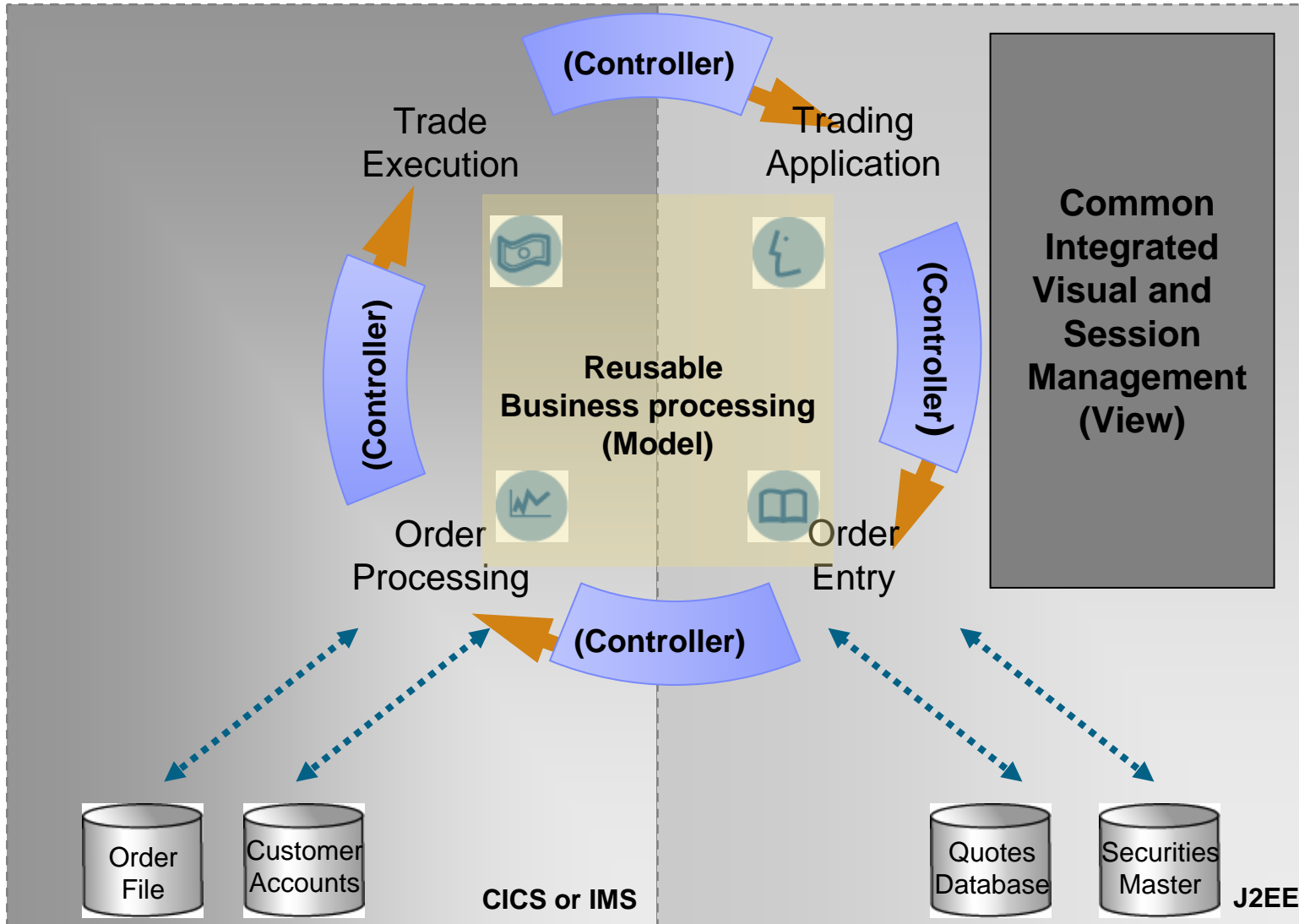
Transform Application
Architecture



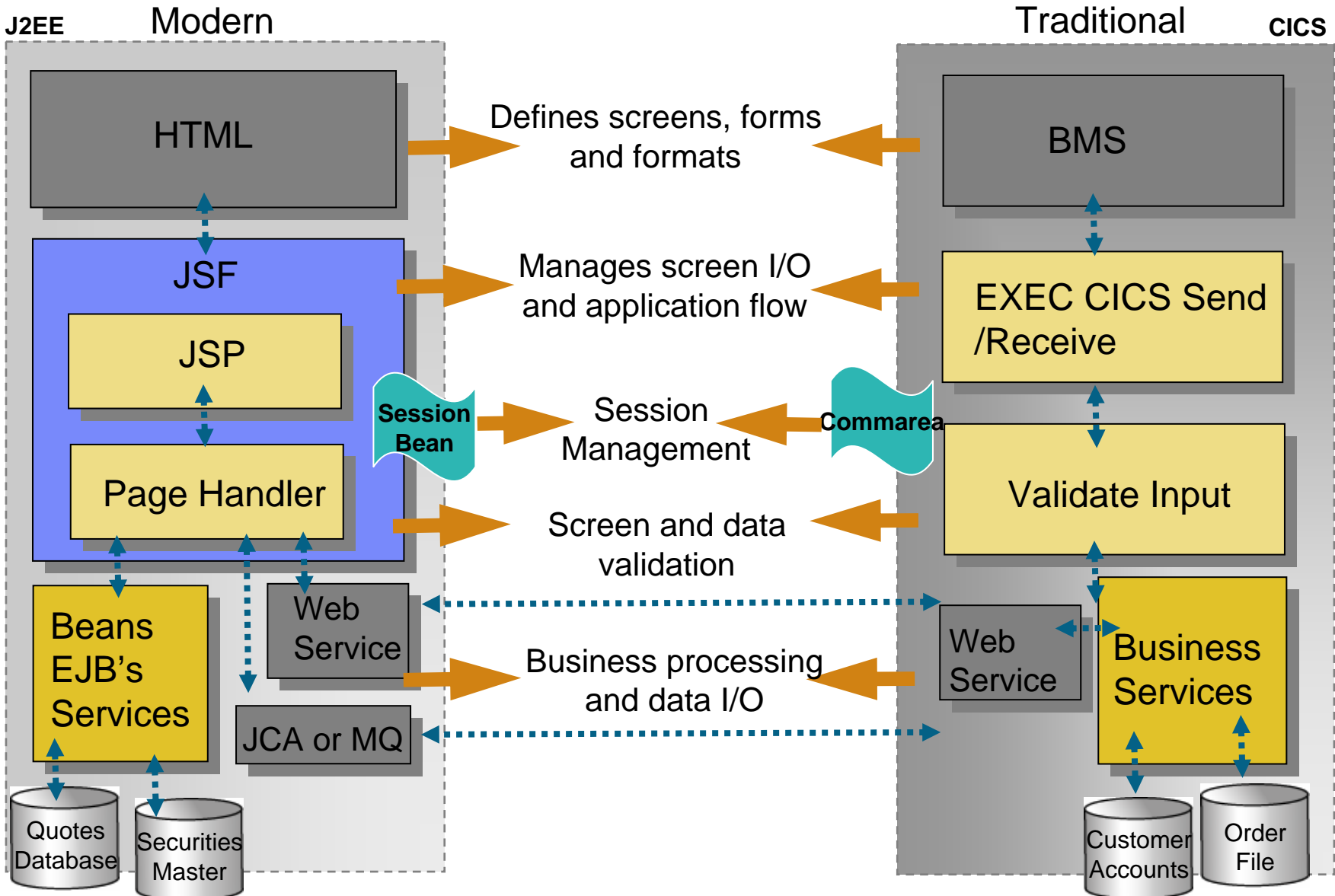
Single integrated delivery vehicle across application transformation styles



Composite Workload *Application Components*



It's not that different



Investment Challenges

3270

COBOL/PL1

ISPF

- **Many zSeries developers still:**
 - Focused on creating or enhancing 3270 applications
 - Using traditional, host-based development environment

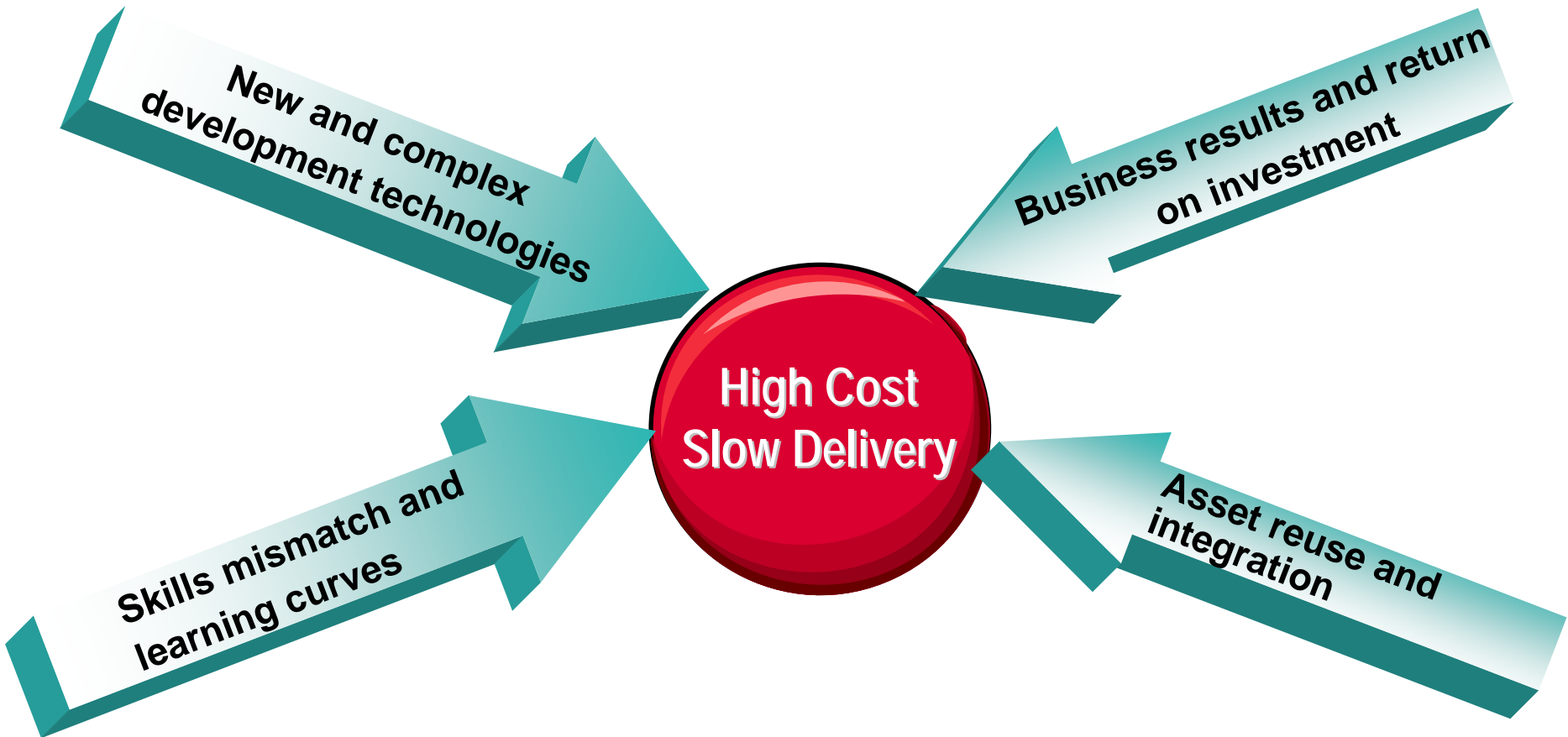
“Application maintenance consumes between 60 – 80 percent of IT budgets” - Phil Murphy, Forrester

Issues: How do I?

- Increase productivity of business developers working on traditional applications
- Enabling broad business developer community in SOA and Web Based infrastructures
- Improve Time to market and IT responsiveness



Technology Challenges



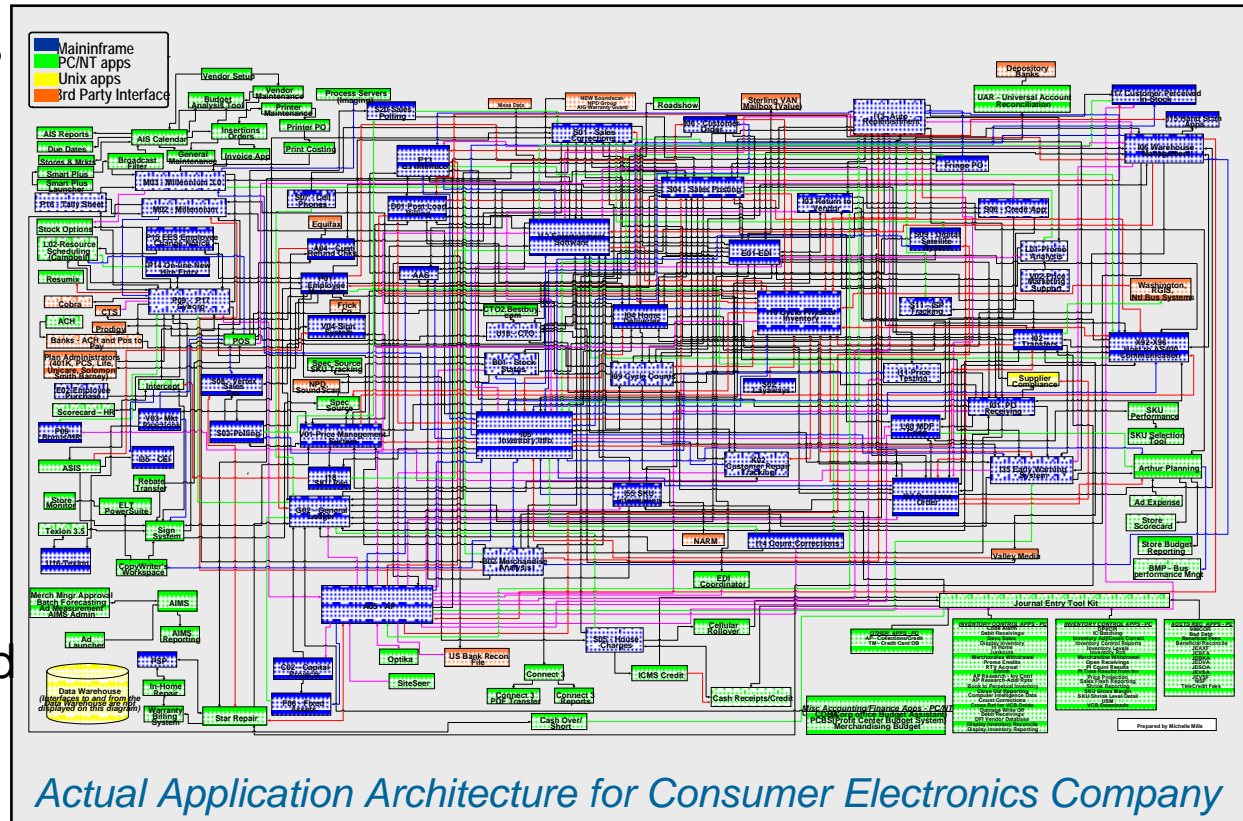
Issues: How do I?

- Enable experts on Core Applications in modern technologies
- Leverage business skills
- Create the SOA infrastructure without throwing everything else away



Architectural Challenges

- Application dependencies are extraordinarily complex, and exist at multiple levels
- Dependencies cross technologies and environments
- Need to support application maintenance, development and test
- Need to support application integration and service / component creation



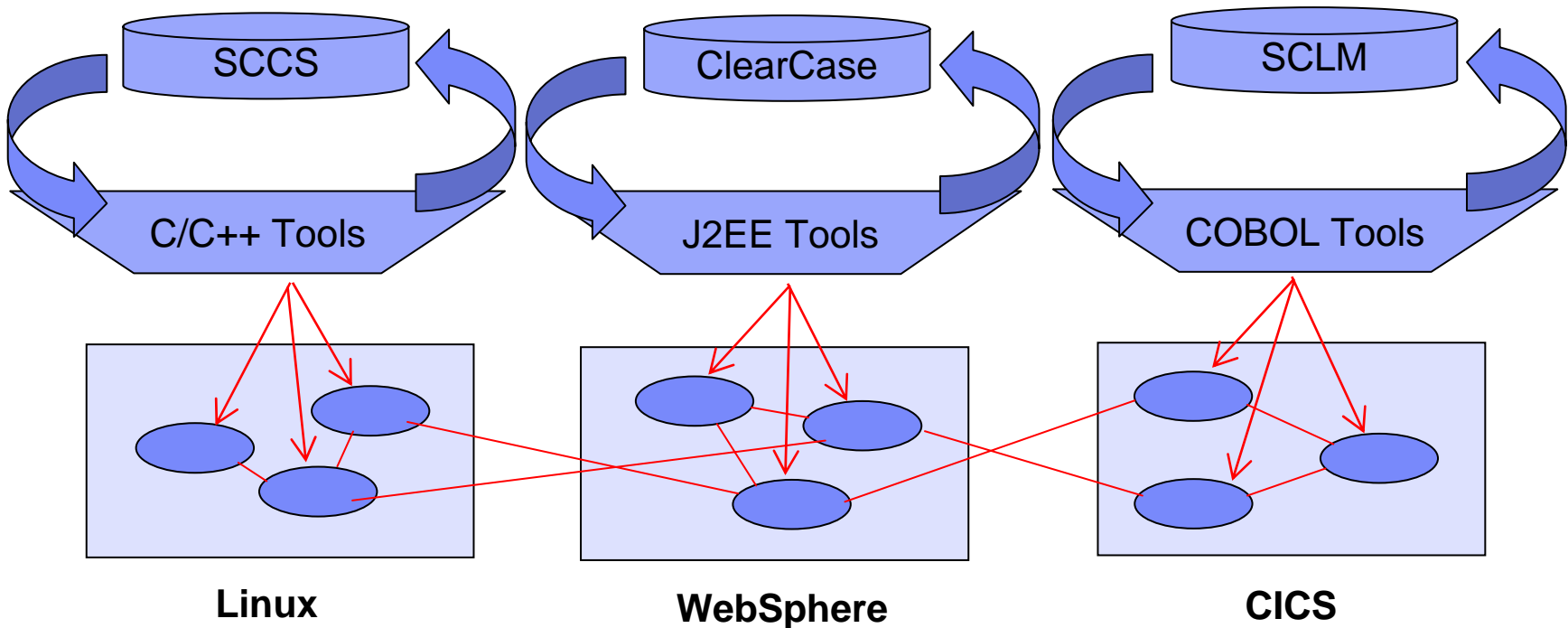
Issues: How do I?

- Improve application backlog and throughput of requirements
- Avoid unplanned impacts – manage quality - during change cycles
- Enable rapid reuse



Organizational Challenges

- Lack application components & skills sharing
- Ineffective / Uncoordinated development of integrated application

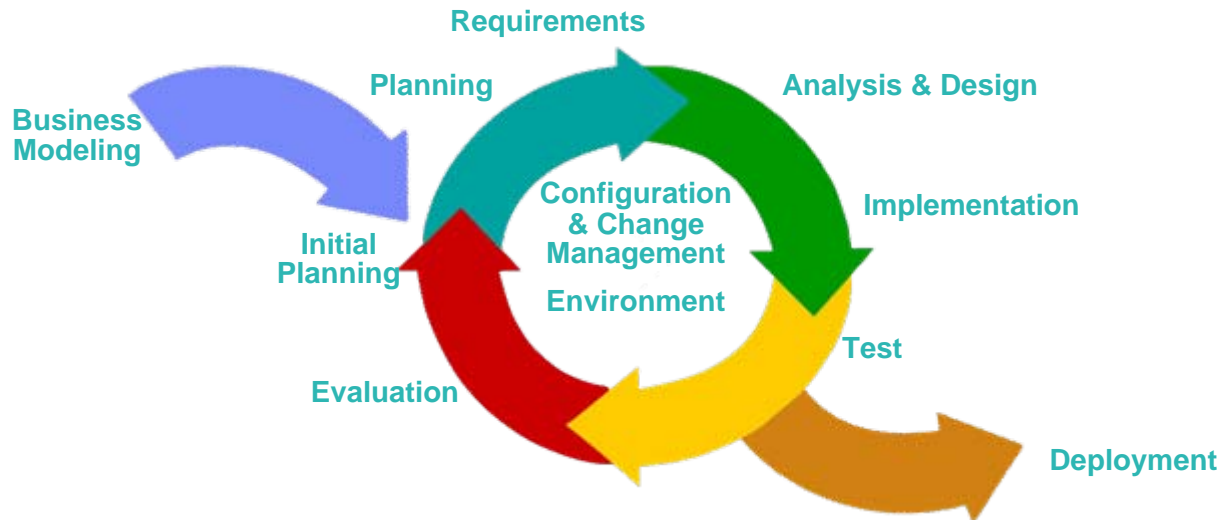


Issues: How do I?

- Manage change across geographically distributed development teams
- Communicate available services and resources
- Leverage existing code – and process – at the same time improving quality



Strategy 1 - Bring iterative model driven development paradigms to composite applications



- Adopt a flexible process for both J2EE & traditional z/Series applications
- Tools integration across the lifecycle (Model and Discover, Develop and Assemble, & Deploy and Manage)
- Manage mixed workload requirements

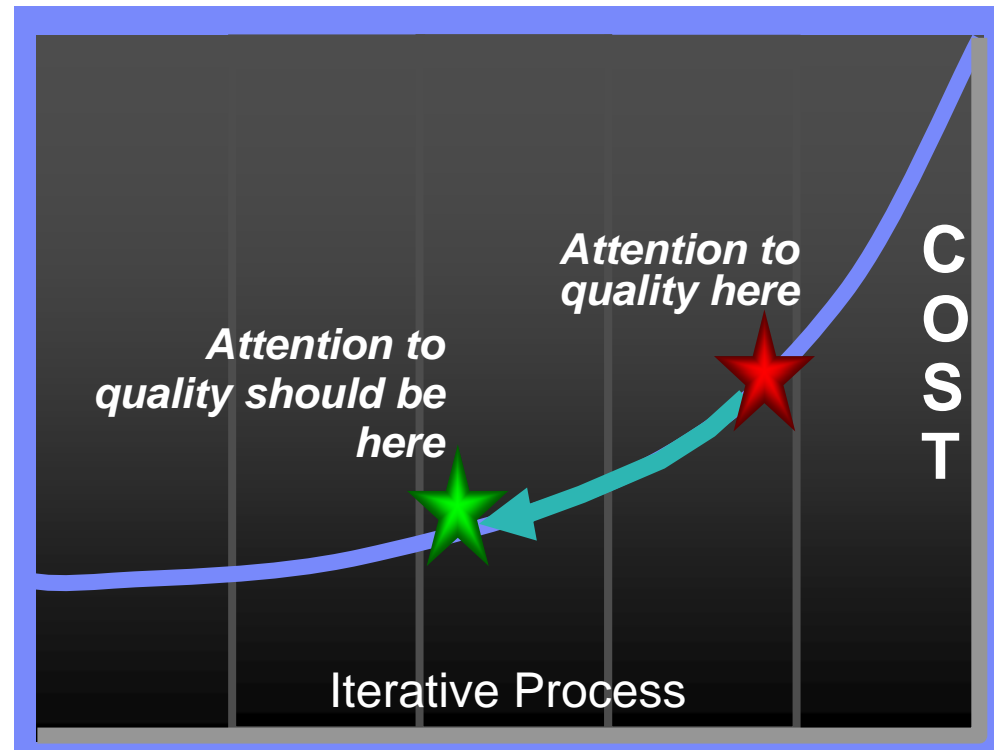
Issues: How do I?

- Leverage modern development techniques across broad developer organizations
- Generate complex SOA architectures, versus hand coding
- Improve documentation and speed the development to test cycle



Strategy 2 - Prevent, detect, diagnose and remove defects

- Improve application quality and test process
- Provide early warnings of activities susceptible to failure
- Analyze across disciplines to understand root causes



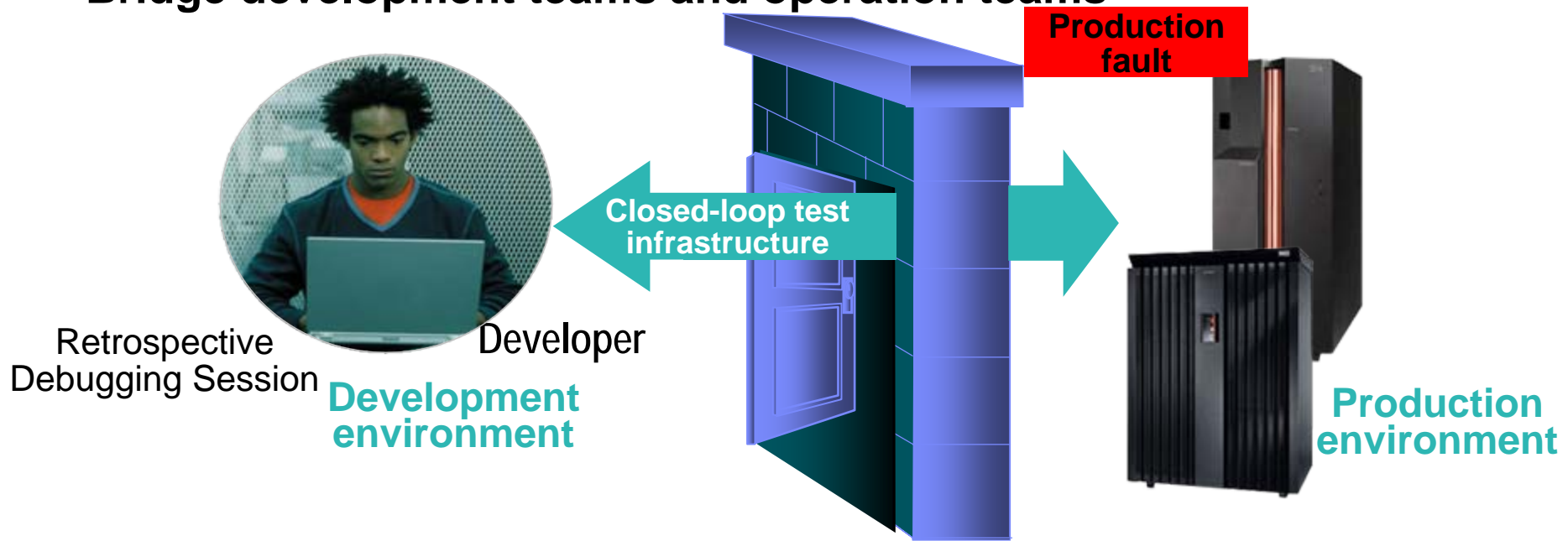
Issues: How do I?

- Find problems in development, before system test and production
- Debug SOA applications cross programs, platforms, languages, etc.
- Perform risk analysis on quality of deliverables



Strategy 3 - Reduce application downtime

- Find and fix errors post-deployment quickly
- Speed application rebuild and redeploy
- Bridge development teams and operation teams



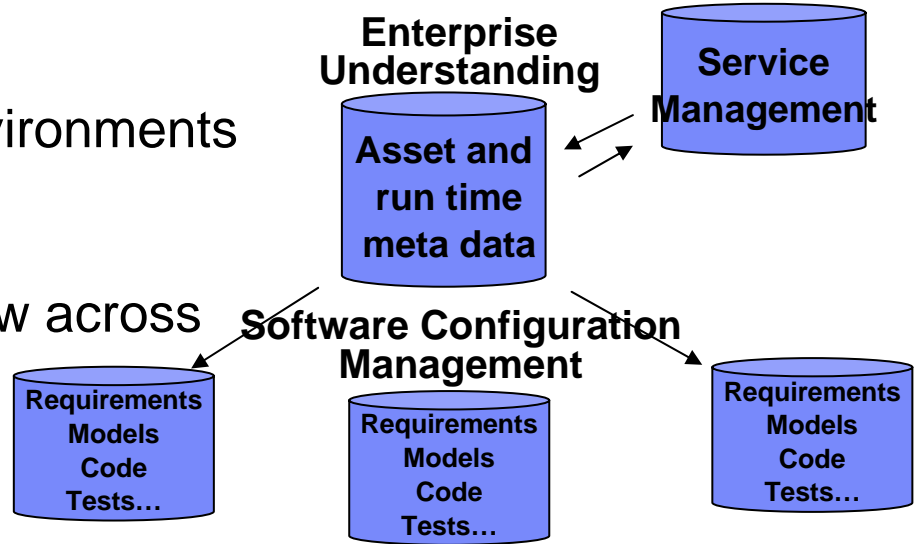
Issues: How do I?

- Manage quality in a SOA environment
- Solve application faults when multiple runtimes are involved
- Leverage business knowledge during problem determination process – i.e., common skills across developer bases



Strategy 4 - Manage change and assets as services

- Manage change across multiple development and operational environments
- Manage diverse assets
- Automate and accelerate workflow across multiple development teams



Business Benefits

- Quickly respond to change
- Develop anytime, anywhere, in parallel
- Enable reuse and protect assets



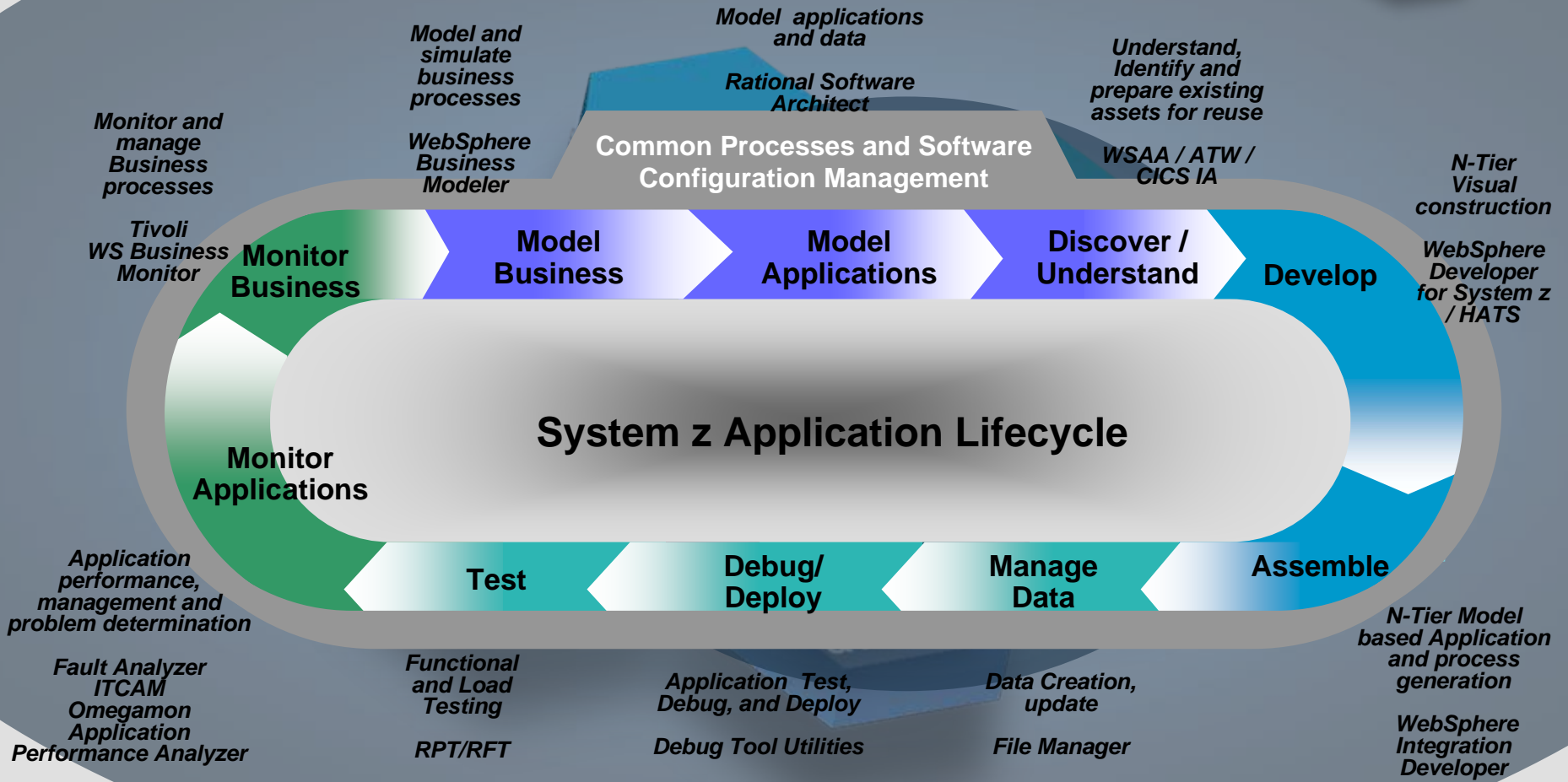
Technology Benefits

- Flexible workflow and process support
- Distributed team management
- Traceability across the lifecycle

Issues: How do I?

- Govern processes and enable reuse
- Track who is working on what
- Merge changes from multiple teams
- Support vastly increased numbers of artifacts across the lifecycle

System z Application Lifecycle



Enabling a robust, flexible SOA runtime environment

While maximizing the value of existing assets

Fully SOA capable!

WebSphere Application Server V6

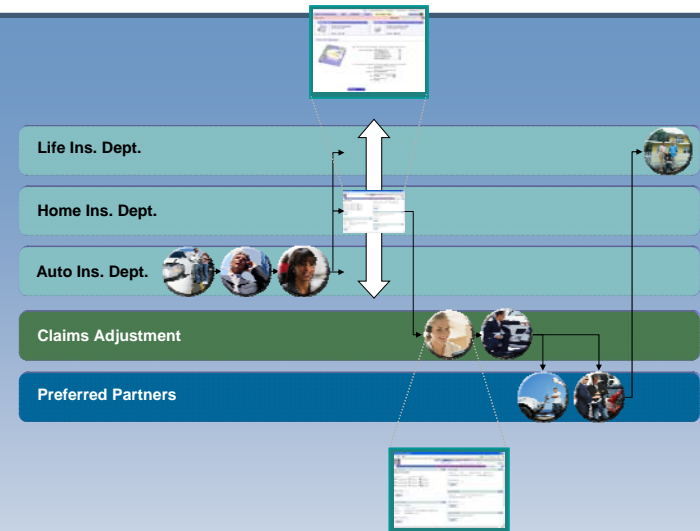
- Extend existing Java assets with support for Web Services standards and standards-based messaging
- Help ensure 24x7 availability of business-critical applications with clustering and high availability
- Build and deploy Web Services quickly and easily with rapid development and deployment features

CICS Transaction Server V3.1

- Exploit provider/requestor Web service support for CICS assets, based on full Web service standards
- Extend the value of CICS transactions in a mixed language environment
- Build Web services from CICS transactions with no change to existing applications.

IMS Transaction and Database V9

- Exploit Web service support for IMS assets, based on full Web service standards
- Extend the value of IMS transactions in a mixed language environment
- Build Web services from IMS transactions with no change to existing applications



#1 in market share for Application Server software

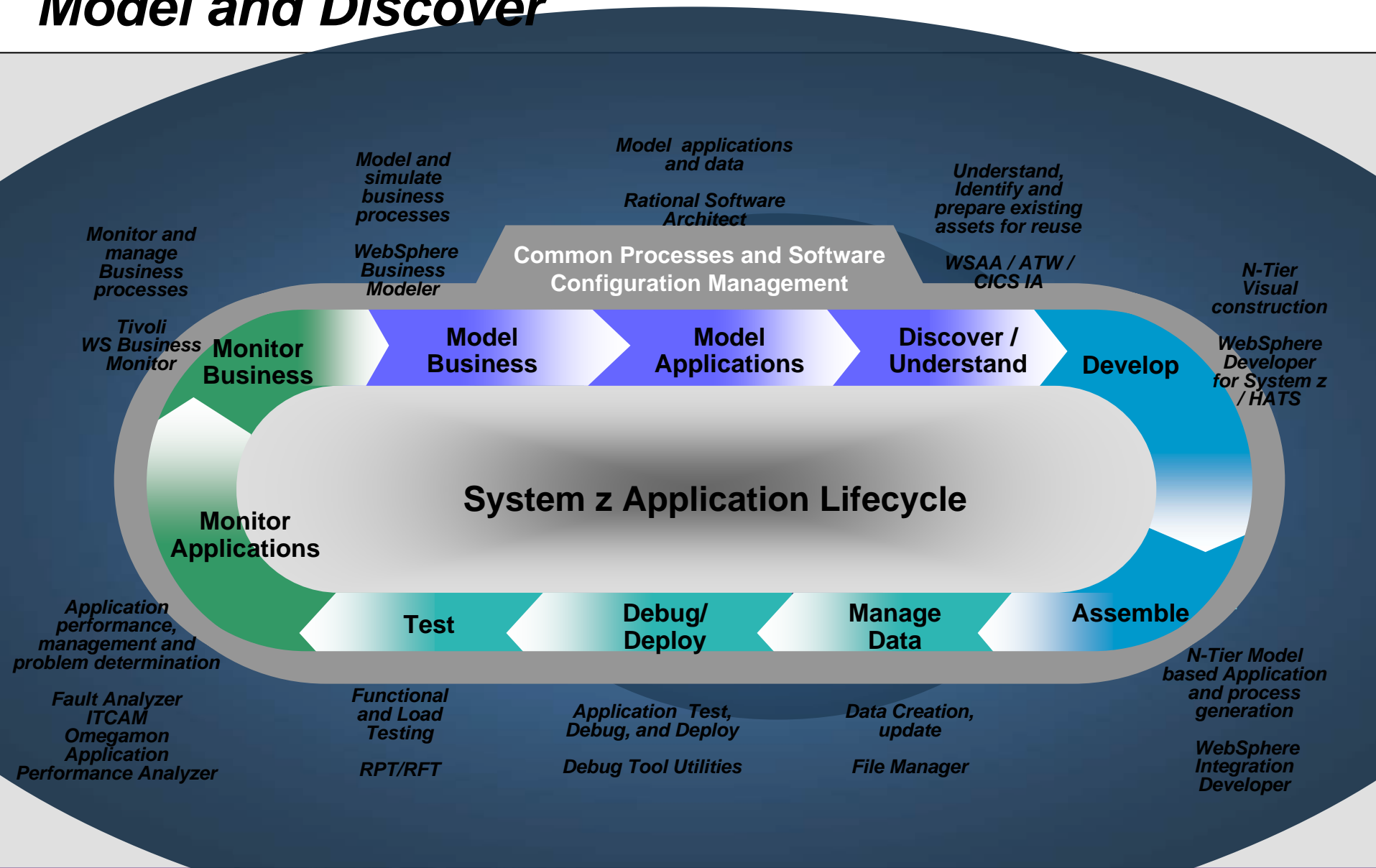


IBM WebSphere Application Server comes out on top

35+ years of maturity and innovation in transaction and data systems



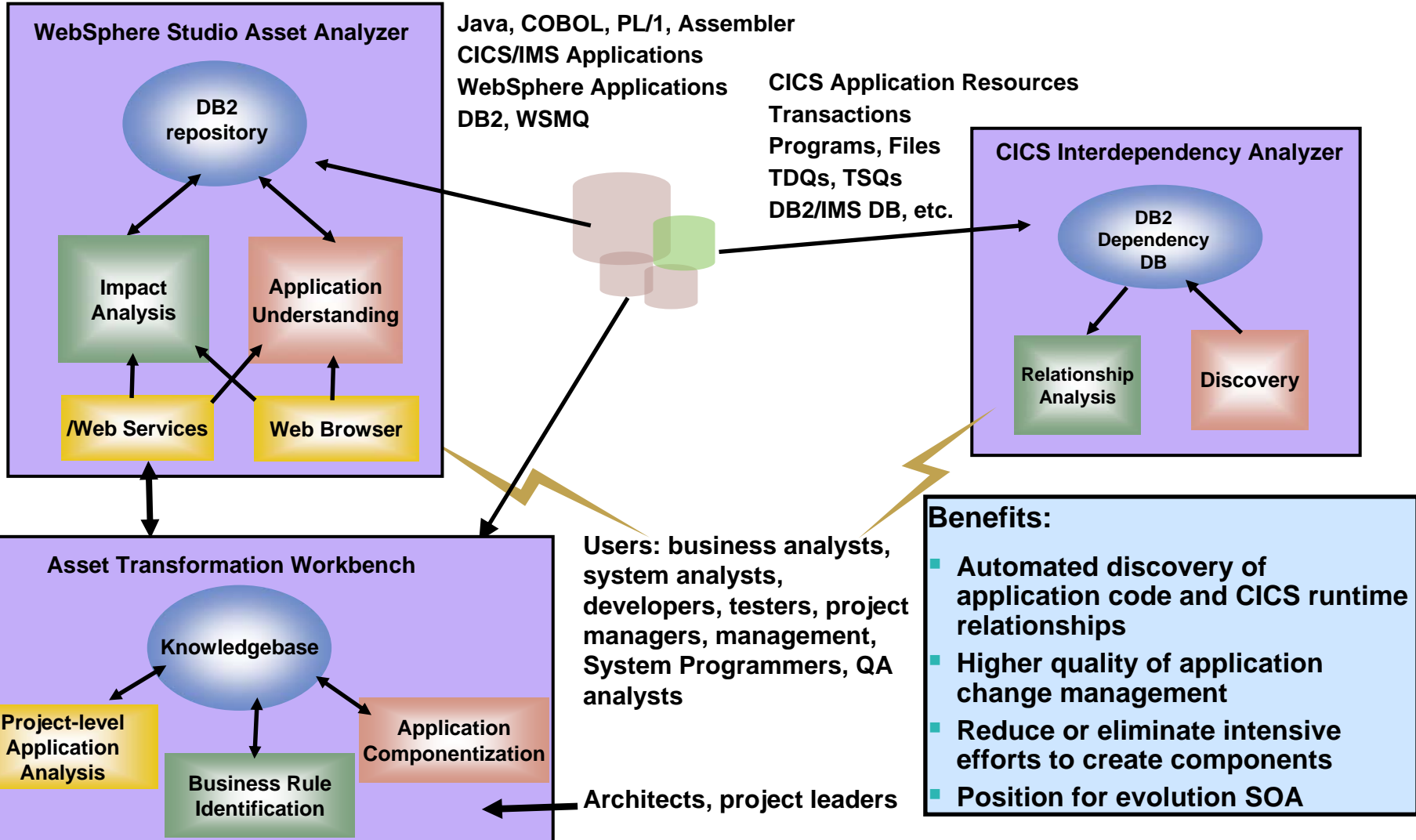
Model and Discover



Enterprise Access to Assets

Speed application discovery, understanding and asset reuse

Enterprise Customer AD artifacts



Model - For The IT Architect and Developer

Using patterns to speed up the process

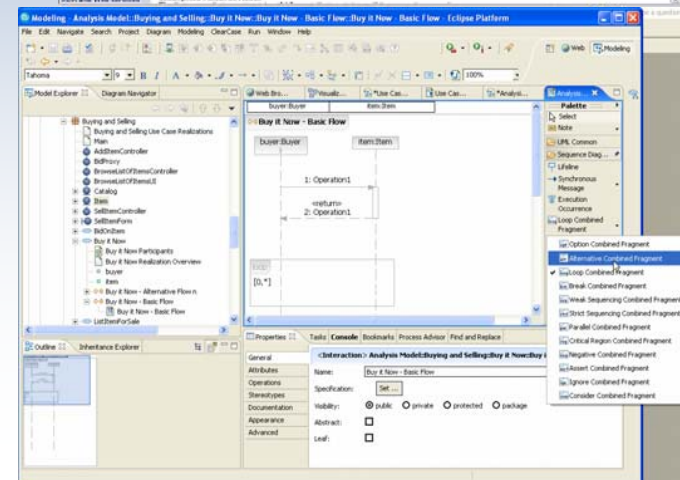
Model using industry standard UML 2, integrating the architecture into development

Rational Software Architect V7.0

- Model in UML and transform to Web service
- Use patterns to help automate development of applications and promote reuse
- Use Process and best practices ensure repeatable success
- Integrates with business process modeling to ensure business needs drive development

Rational Software Architect Pattern Solutions

- Improve productivity with reusable assets
- Rapidly build and configure the Enterprise Service Bus (ESB) with the WebSphere Platform Messaging Patterns



Identify Assets

WebSphere Service Registry and Repository

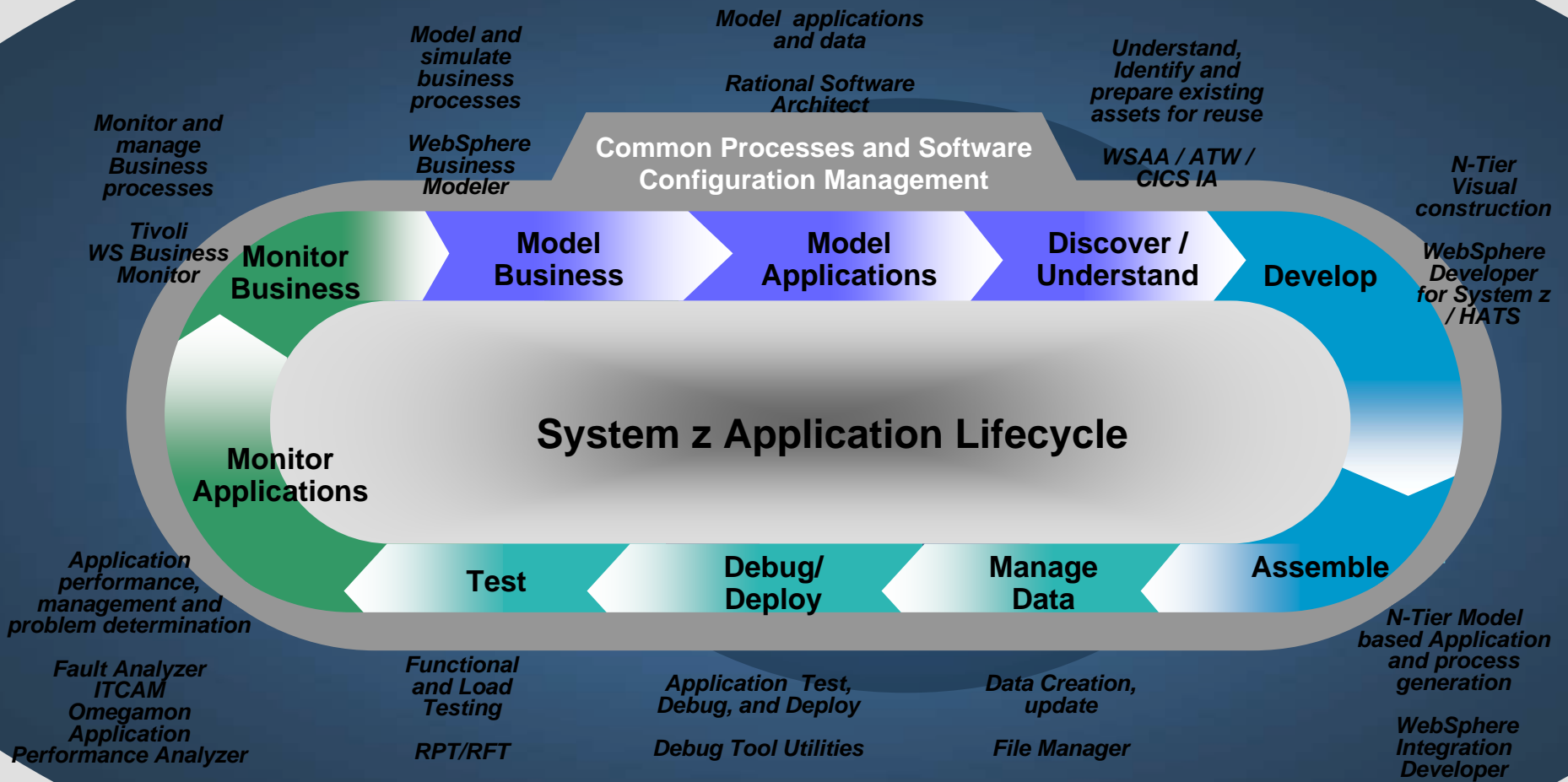
An enterprise-wide service registry and repository improves visibility, reusability, adaptability, and manageability of services

The WebSphere Service Registry and Repository ...

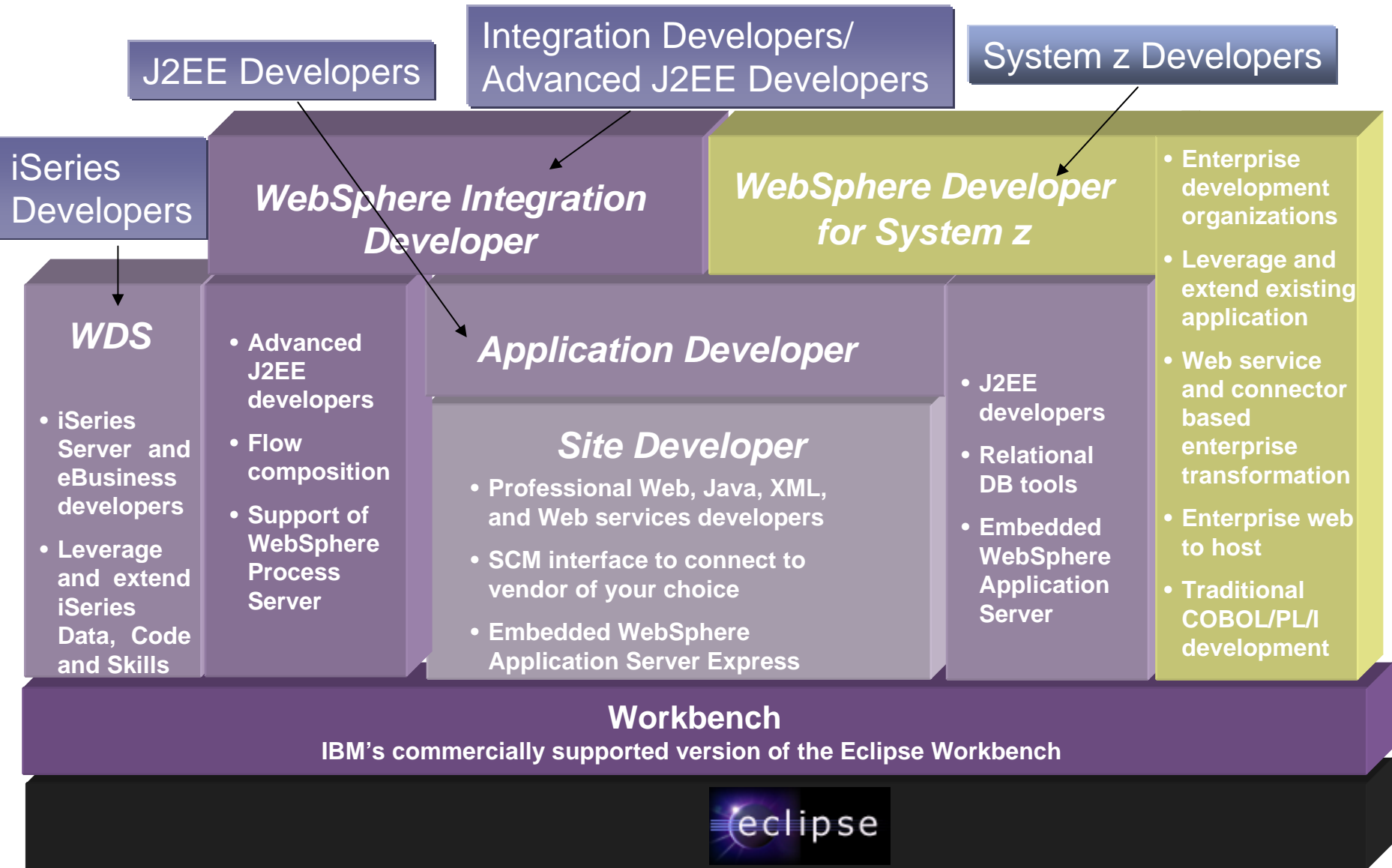
- A **repository** for service metadata
 - for example, WSDL and XSD
- For **publication** of services
 - to advertise their capabilities
- For **finding** suitable services
 - for reuse and runtime agility
- For **capturing** service dependencies
 - to support change management
- An **extensible** framework
 - to support validation and notification



Develop and Assemble



WebSphere/Rational Development Family



IBM WebSphere Developer for System z

Enterprise Service Tools Web Services and XML

- SOA access to CICS V3.1 and IMS V9 COBOL applications
- Bottom-up/Top-down or meet-in-the-middle COBOL to XML mapping support
- Integrated COBOL XML converters, XML schemas, and WSDL generation

DB2 Stored Procedure for COBOL and PL/I

- Create DB2 stored procedures on z/OS in either COBOL or PL/I
- Build and catalog support for the DB2 stored procedure
- Debug z/OS based stored procedures from workstation

Web/J2ee Development

- Create Web Pages / JSF / Struts
- J2ee/Java Development
- JCA Connectors
- Distributed debugger
- Web Services and Test environment



z/OS Application Development

- Connect to z/OS systems
- Work with z/OS resources like COBOL, PL/1 , C, C++, JCL, etc.
- Interact with the Job Entry Subsystem (JES) to submit jobs, monitor jobs, and review job output
- Perform dataset management actions like allocating datasets and migrating datasets
- Perform typical edit, compile, and debug tasks on remote z/OS resources from the workstation

CICS BMS/ IMS MFS Map Support

- Visually create and modify BMS Map sets or IMS/MFS
- Generates JCL
- Work with local or remote maps

Rapid Application Development

- Ability to generate WSDL and CICS Cobol program to access DB2
- CRUD from UML and Database operations
- Can be added to SFM flow

CICS 3 Service Flow Support

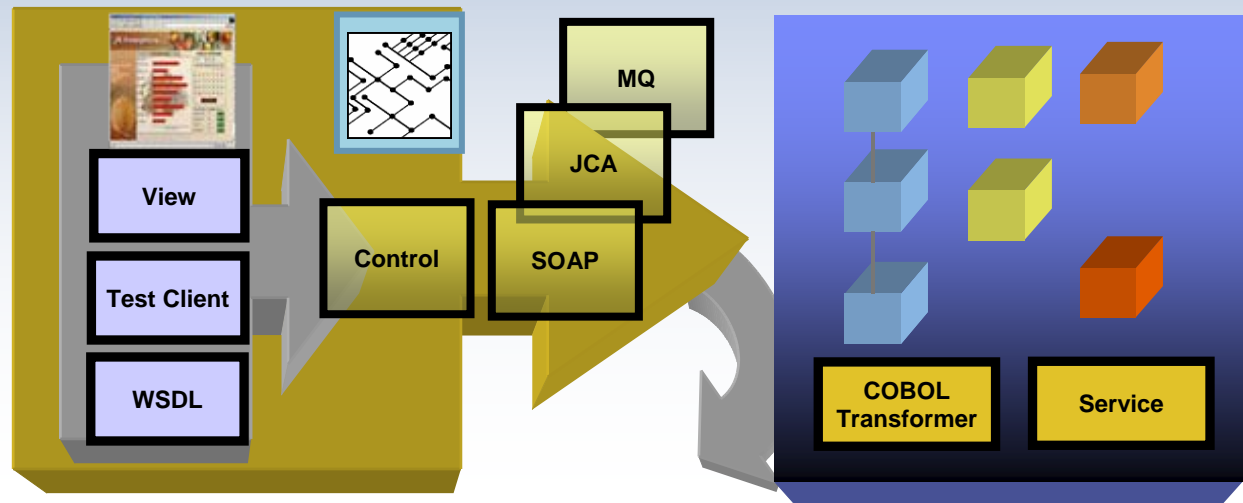
- Implements SOA and Web Services
- Service Flow Modeler is a tool to build service flows out of your existing Commarea and Terminal based CICS applications.



z/OS Composite Development tools

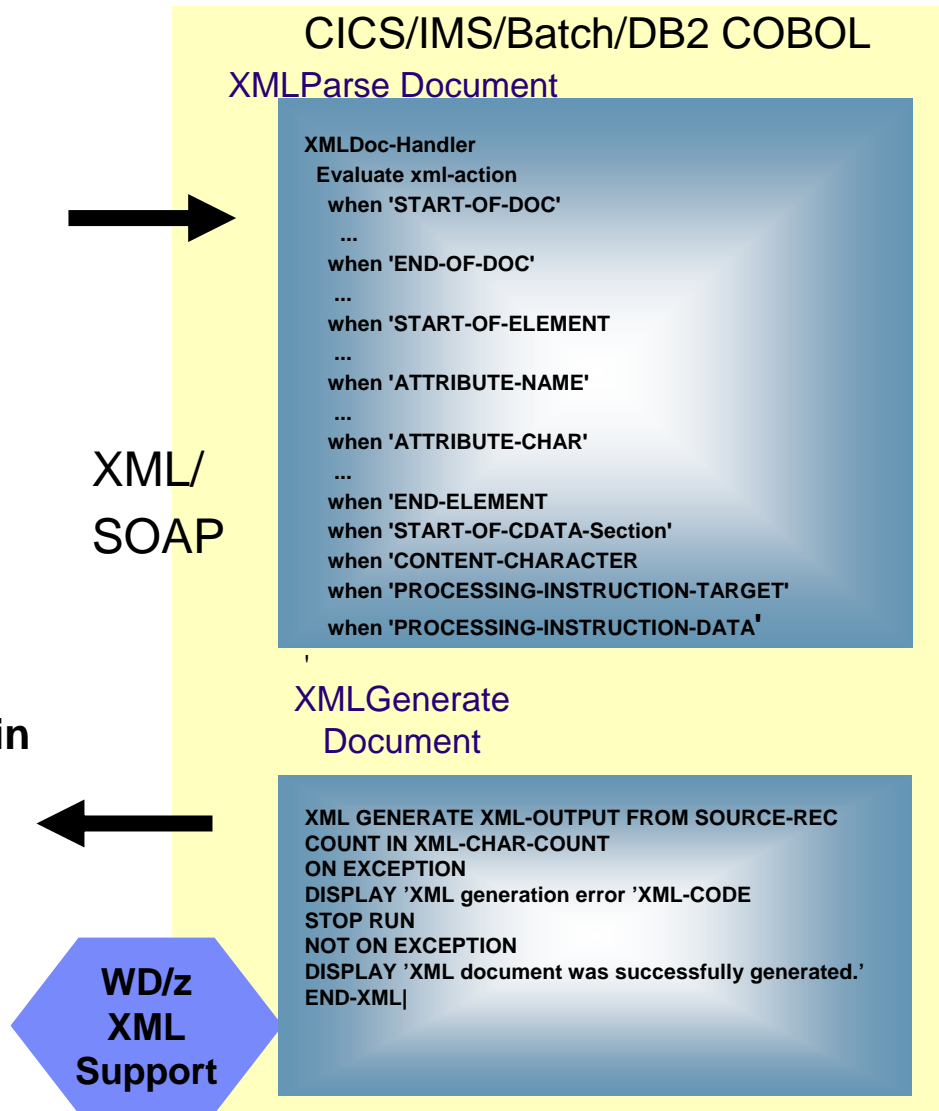
Transition of Traditional environments to Web and Composite applications

- **SOA / SOAP / XML / Enablement**
- **JCA Support**
- **Service Flow Modeler**
- **HATS**
- **Enterprise Generation Language (EGL) / JSF**
 - COBOL/CICS generation
 - Java generation



Using Enterprise COBOL to service-enable z/OS

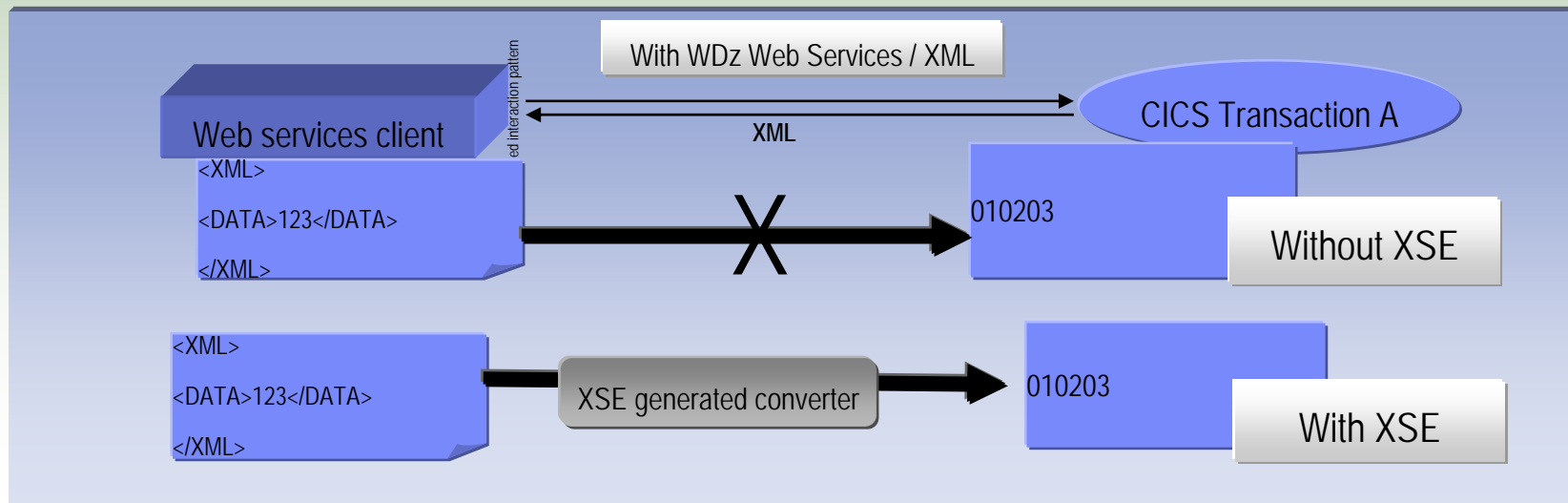
- **What's the latest...**
 - XML Language based generation from COBOL data structure
 - WebSphere EJB support
 - DB2 V8 preprocessor
 - CICS preprocessor
- **High speed XML Sax based parsing**
- **Object Oriented Support for Java COBOL Interoperability**
- **Unicode support**
- **Similar XML parsing support available in Enterprise PL/I**
- **CICS and DB2 integrated preprocessor**
- **Raise 16Mb COBOL data size limit**
 - Picture clause replication:
01 A PIC X(134217727).
 - OCCURS::
05 V PIC X OCCURS 134217727 TIMES.



WebSphere Developer for system z SOA Tools – Part 1

Enterprise Service Tools

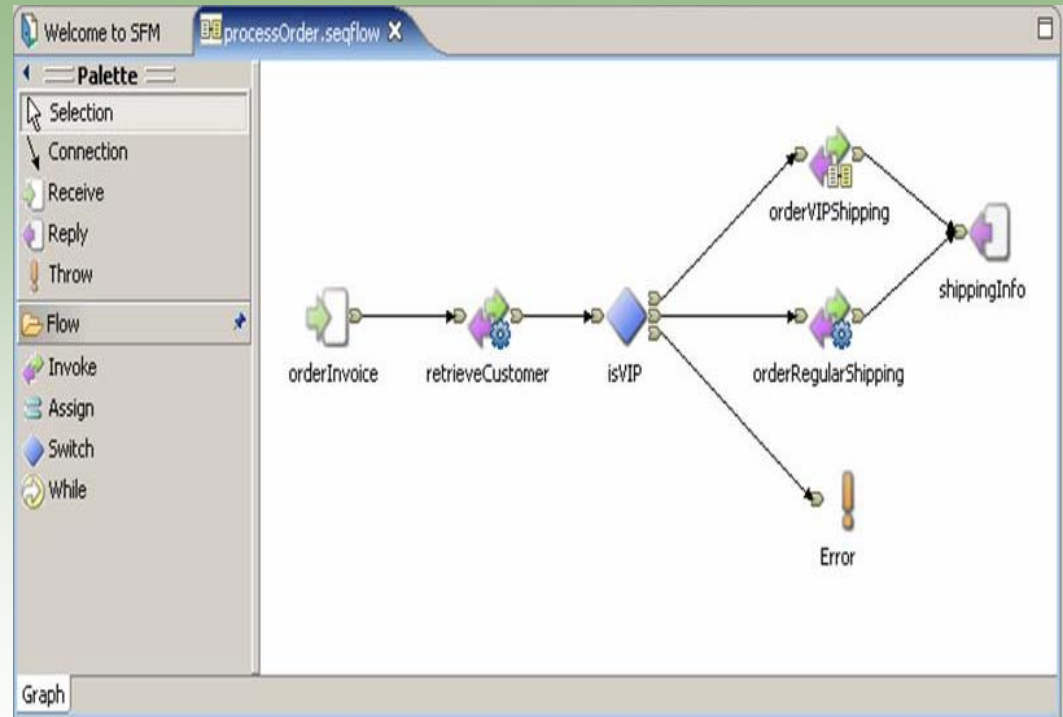
- Most rapid building of Web services from existing CICS applications
 - Single CICS and IMS transactions enabled for Web Services
 - Supports IMS Message Queue, CICS Commarea and new Channels/Container based applications
 - Rapid generation of WSDL, CICS WSBind, and Adapter generation eliminating complex hand coding of XML to/from language conversions
 - Includes complete Web Services Test and Java generation environment



WebSphere Developer for System z SOA Tools – Part 2

Enterprise Service Tools - Service Flow Modeler

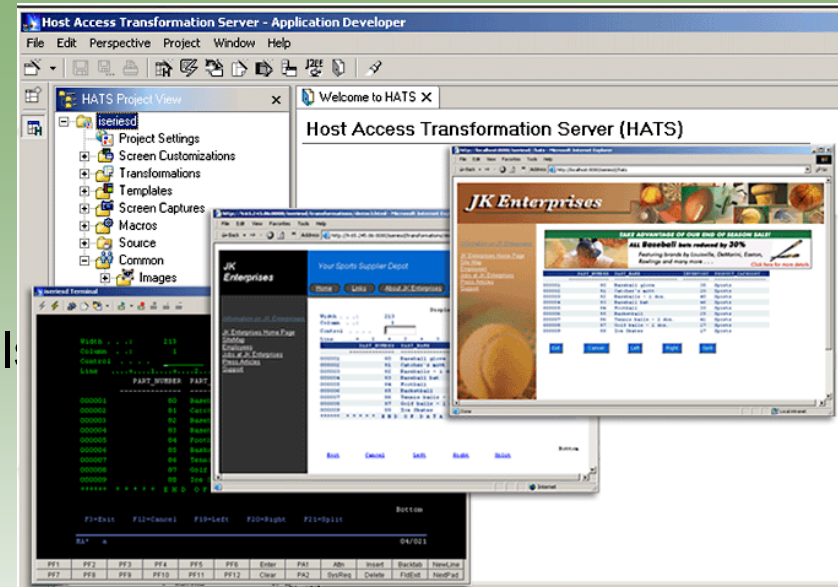
- **Builds Web services from existing CICS applications**
 - Aggregates multiple CICS transactions into high-level business processes through visual modeling
 - Supports CICS BMS (terminal-based) applications & CICS commarea applications
 - Highly optimized CICS runtime supporting Web services and XML interfaces



WebSphere Host Access Transformation Server

Extend business processing through existing interfaces

- Automatically transforms 3270 & 5250 green screen applications into HTML interfaces
- Extends terminal applications as Web Services
- Low skills requirement – no System z skills required
- Rules-based, highly customizable
- Iterative, eclipse-based development environment

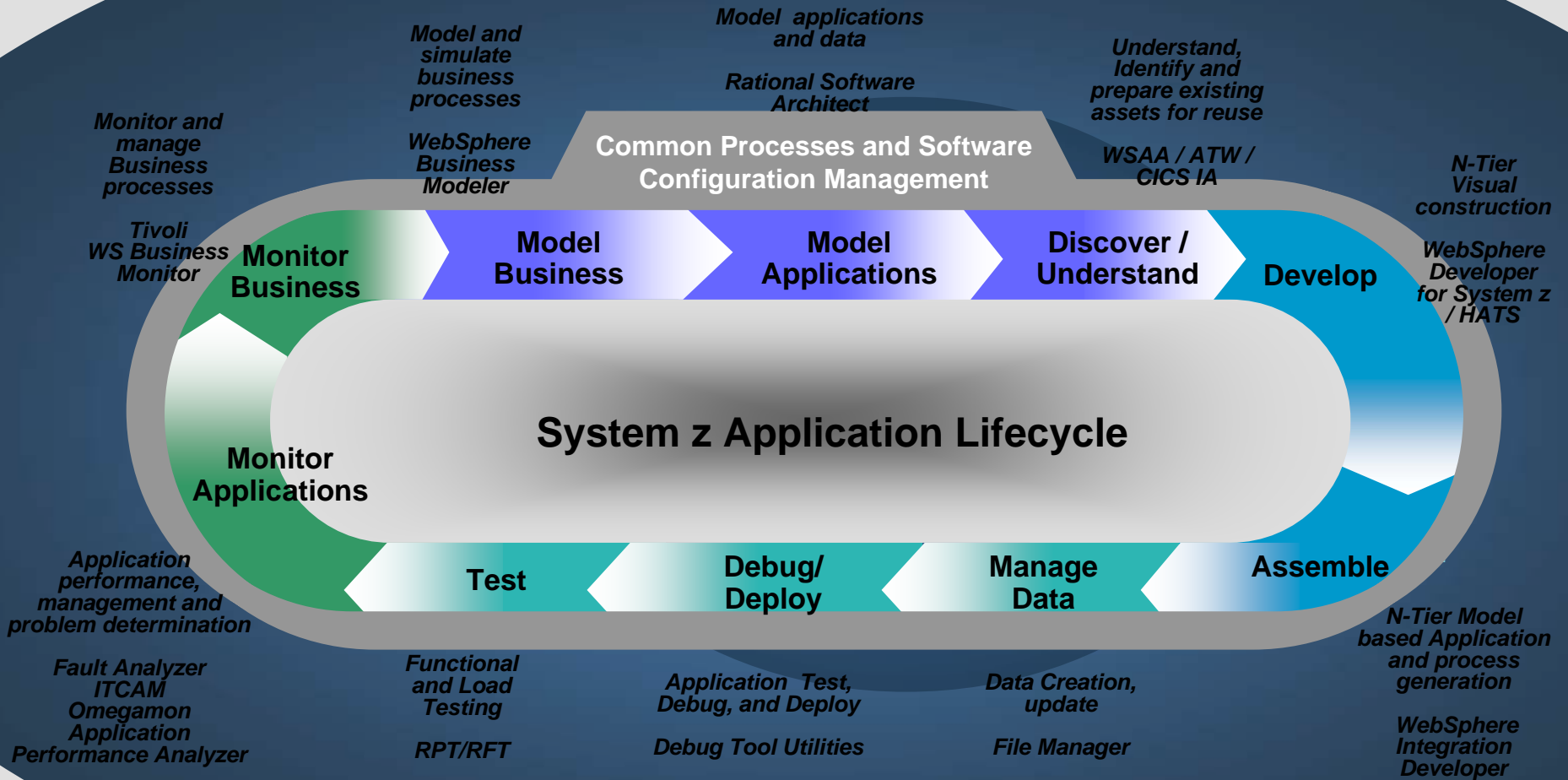


Benefits:

- Increase productivity and reduce training costs.
- Extend existing applications to new users
- Integrate traditional applications into enterprise portals
- Reduce development costs by avoiding rewrite of legacy applications.

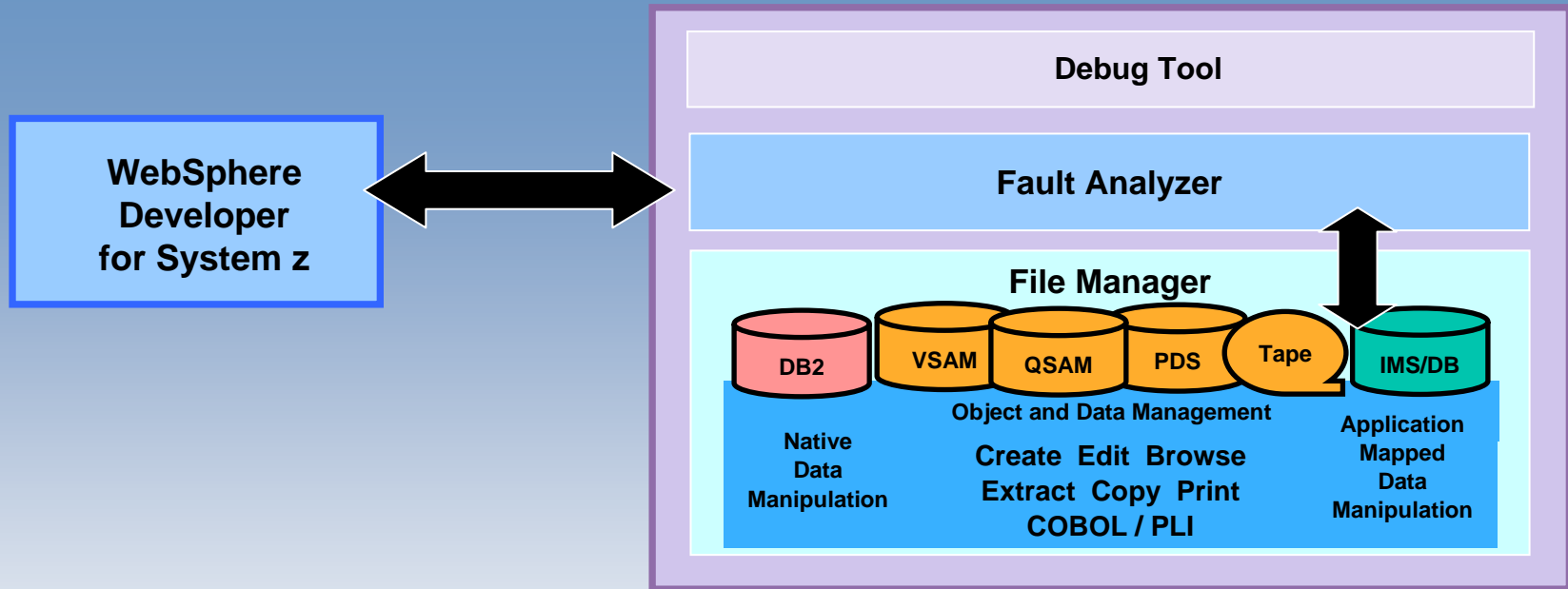


Deploy and Manage



Test and Problem Determination

Integration speeds time to market



Benefits:

- Simplify development of System z test cases
 - Data creation for DB2, IMS/DB, VSAM, and QSAM
 - Extract and load
- Reduced deployment complexity
 - Production data validation and creation
- Common environment
 - Reuse of skills across e-bus and traditional applications

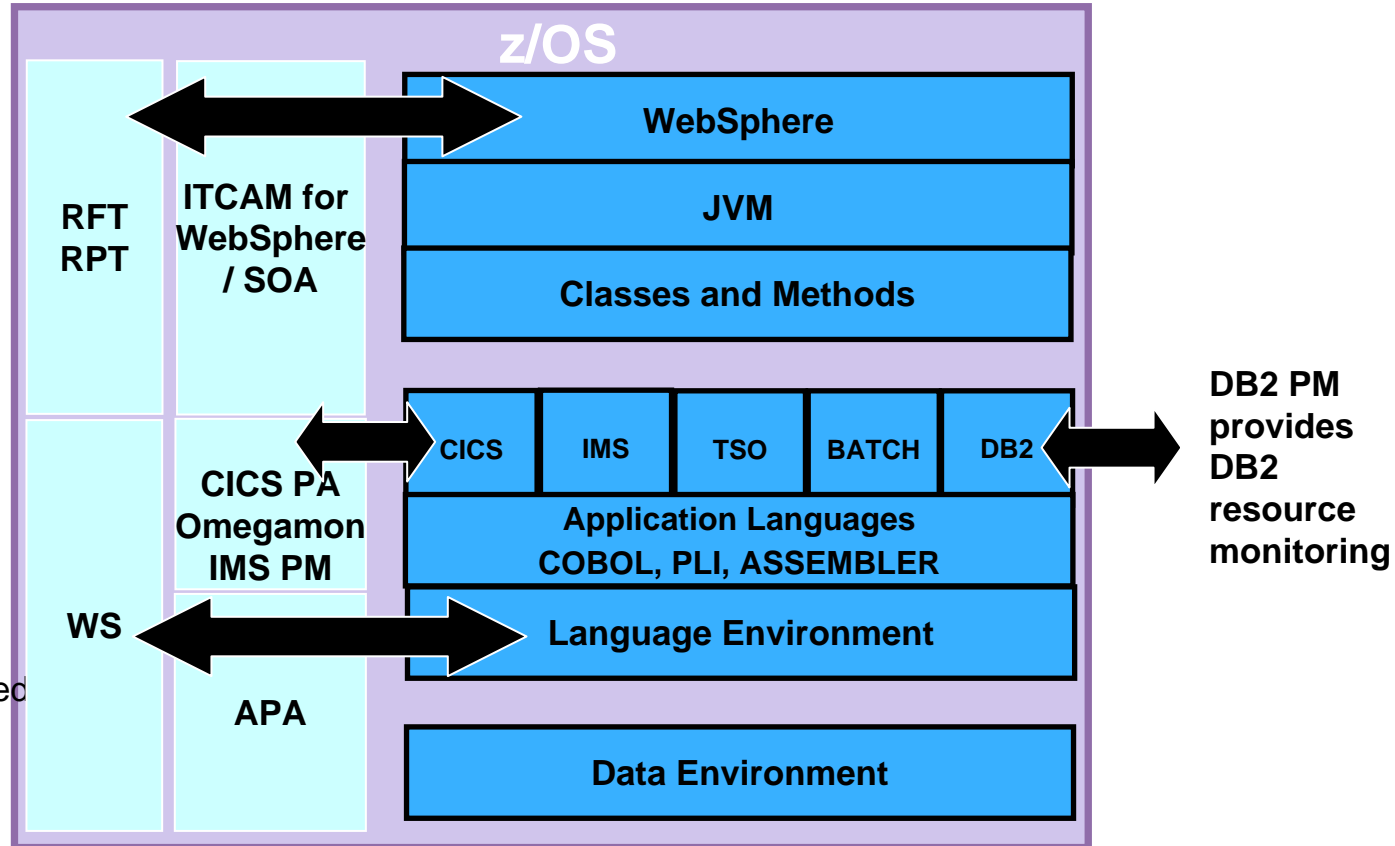


End To End Monitoring

Enables highest QOS and maintainability of composite applications

Benefits:

- RPT, ITCAM used to drive and monitor J2EE performance on both WAS and traditional servers enabling rapid problem determination and reduced downtime
- CICS PA / OMEGAMON provide CICS and IMS resource monitoring enabling rapid response to problems
- System z WS and PA are used to drive and monitor CICS transactions and DB2 performance for COBOL / PLI applications enabling high throughput in System z environments



Deploying processes on a flexible, robust SOA integration platform

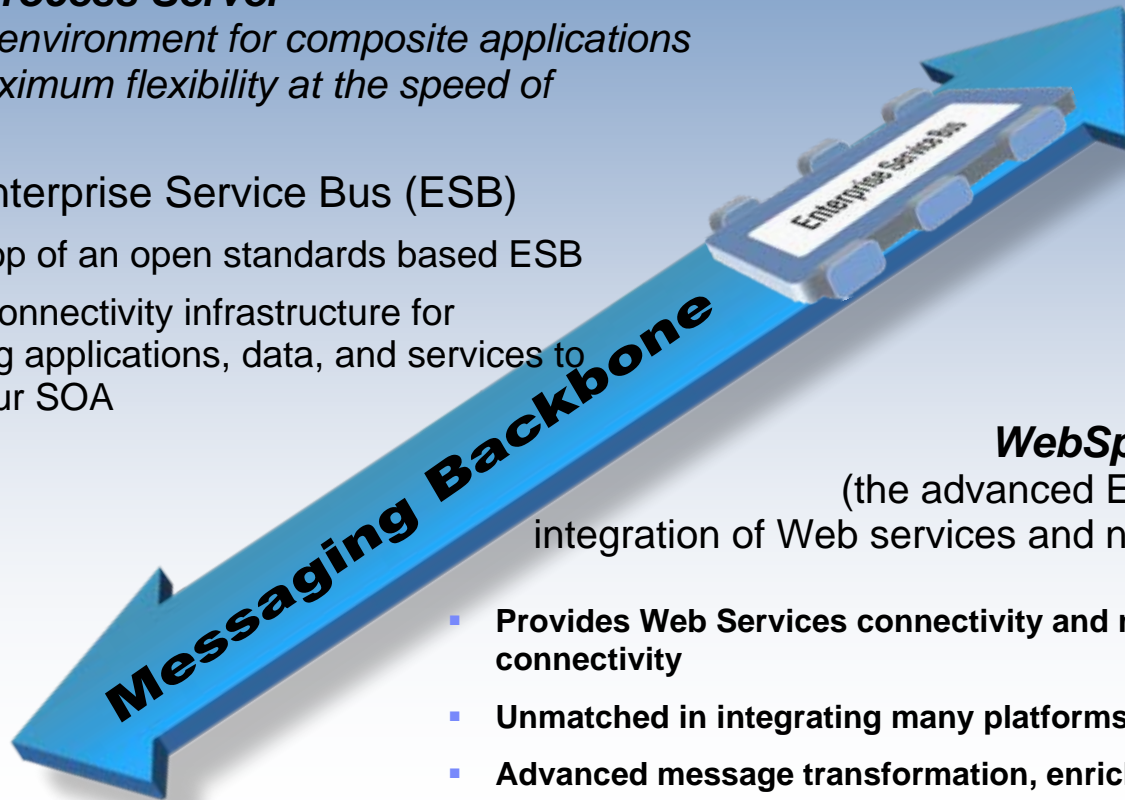
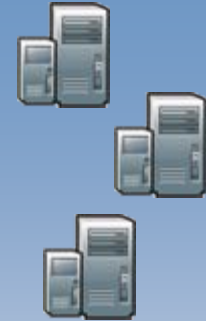
Employing mediation to enable every kind of application and data –to participate in SOA

WebSphere Process Server

(A deployment environment for composite applications to ensure maximum flexibility at the speed of business)

Powered by Enterprise Service Bus (ESB)

- Built on top of an open standards based ESB
- Flexible connectivity infrastructure for integrating applications, data, and services to power your SOA



WebSphere Message Broker
 (the advanced ESB for high performance integration of Web services and non-Web services assets)

- Provides Web Services connectivity and non standard interface connectivity
- Unmatched in integrating many platforms, devices, and APIs
- Advanced message transformation, enrichment, and routing



Gartner: Best Practices for Mainframe SOA

- **Act tactical, think strategic**
- **Evaluate tools that provide good microflow orchestration**
- **Create services that utilize function from across existing application boundaries.**
- **Build a reuse culture and technology infrastructure.**
- **Work with operations to create management/performance-monitoring support.**
- **Use code understanding/inventory/restructuring tools to improve service granularity.**
- **Define the role of the mainframe in future application architecture.**

