



IBM Software Group

z/OS Application Transformation in SOA environment

Service Oriented Architecture

An IBM Exploration of Technology



Agenda

- **Introduction to Application Transformation and SOA** (30 min)
 - ▶ Application Transformation in the SOA environment
 - ▶ IBM tools to transform existing applications in the SOA architecture
 - ▶ Scenarios introduction
- **Scenario #1 - Creating of a Web Screen from existing terminal based CICS application**
 - ▶ WSAA introduction .. Why do I need WSAA ?
 - ▶ HATS introduction .. Why you need HAT?
 - ▶ **Demo #1** – Considering the above scenario shows the Web Page creation, use **WSAA** to find the screen and the components, use **HATS** to create/deploy the Web Page, run using WebSphere Application Server
- **Scenario #2 - Transform an existing program COBOL and create a called subroutine to isolate the business logic**
 - ▶ WSAA Bridge and ATW introduction (15 min) ..
 - ▶ **Demo # 2** - (15 min) – Use **ATW** to extract the business logic and create a COBOL subroutine to be used later.
- **Wrap-up** -

Introduction to Application Transformation and SOA

- Business requirements
- I/T Challenges
- Service Oriented Architecture
 - ▶ Introduction
 - ▶ Challenges for System z Customers
 - ▶ Strategies
- SOA and the System z Application Lifecycle



Current business requirements..



*Many roads
lead to ...?*

- Reducing **time-to-market** for new products and services
- Shifting to a **customer-driven-philosophy**
- Streamlining **transaction flow across various business areas**
- Creating **flexible information systems** to archive business agility
- Adapting systems to **changing business processes**

I/T challenges

- IT Imperatives
 - ▶ Support growth
 - Improve flexibility and responsiveness
 - ▶ Keep costs in check
 - Do more with less
- Driving Need to Transform and Integrate Existing Applications
 - ▶ Extend **existing** applications to new audiences and opportunities
 - ▶ Exploit **existing** resources and skills
 - ▶ Improve **performance** of existing workloads for faster **response** times and **reduced costs**
 - ▶ Improve **system management** to enable management of more with less
 - ▶ **Simplify** the **development** process to reduce application development costs and time to deployment

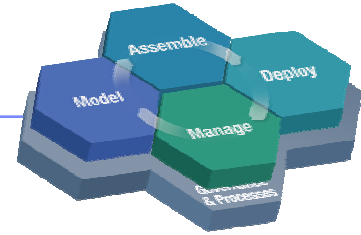
Responsiveness: the new key competence

So growth is back on the agenda – but it won't just happen by itself. CEOs all over the world have identified organizational responsiveness, agility and flexibility as necessary competencies. Developing the ability of the organization to not just sense, but to anticipate and respond to the changing marketplace and subsequent customer requirements is one of the great challenges for today's CEO.

CEOs are now focusing on how their organizations read, listen and react to dynamically changing external and internal conditions. As one CEO put it, "we have to implement a competitive

Source: IBM Global CEO Survey, Feb 2004

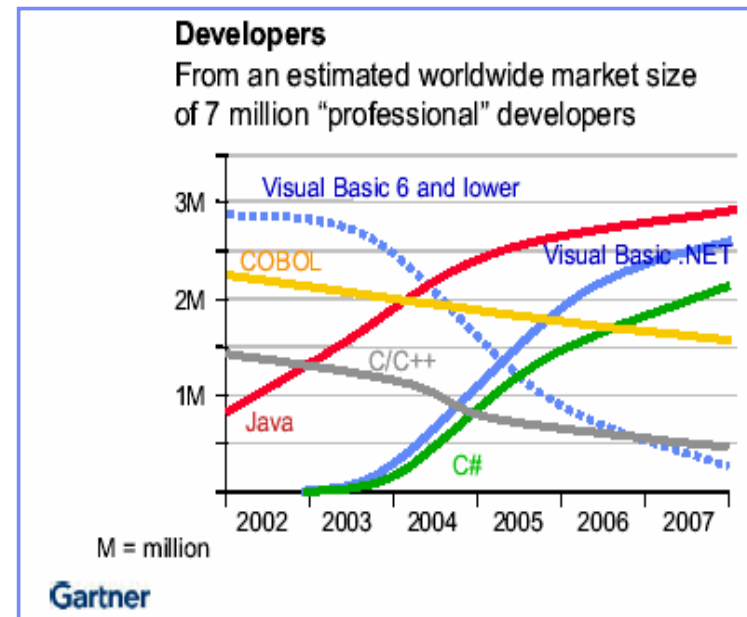
Legacy applications



- 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?



A simple Google search..

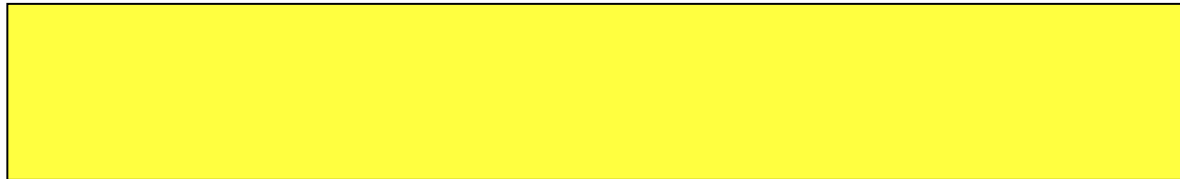
<http://www.cioinsight.com/article2/0,1397,764165,00.asp>

RESEARCH

December 13, 2002

Legacy Systems 2002: Are Your Older Systems Slowing You Down?

According to this month's *CIO Insight* research, corporate America is looking to migrate from legacy systems—in its own good time.



<http://www.cobolportal.com/developer/detart.asp?cat=business&bhcp=1>

Network World
By John Dix
20 October 2003

A fair criticism of the trade press is that we are fascinated with new technology and lose sight of the fact that you folks are mired in legacy molasses. Take COBOL. While headlines today focus on things such as Java and Microsoft's .Net according to statistics gathered by The Senior Staff a San Jose databank of IT professionals over the age of 35: • 75% of business data is processed in COBOL (Source: Gartner). • There are 180 billion to 200 billion lines of COBOL in use worldwide (Gartner). • 15% of new applications are written in COBOL (Gartner). • Replacement costs for COBOL systems estimated at \$25 per line are in the hundreds of billions of dollars (Tactical Strategy Group). [more >](#)

The legacy systems are still relevant and reliable

A simple Google search..

<http://cwflyris.computerworld.com/t/915781/281087/36897/2/>

Cobol Coders: Going, Going, Gone?

Many IT managers slowly migrating away from Cobol wonder whether they'll run out of Cobol programmers before they run out of Cobol code. By Gary Anthes

Gary Anthes [Today's Top Stories](#) > or [Other Development Stories](#) >

October 09, 2006 (Computerworld) -- Cobol, that mainstay of business programming throughout the 1960s, '70s and '80s, is not going away anytime soon. In a *Computerworld* survey early this year of IT managers at 352 companies, 62% of the respondents reported that they actively use Cobol. Of those, three quarters said they use it "a lot" and 58% said they're using it to develop new applications.

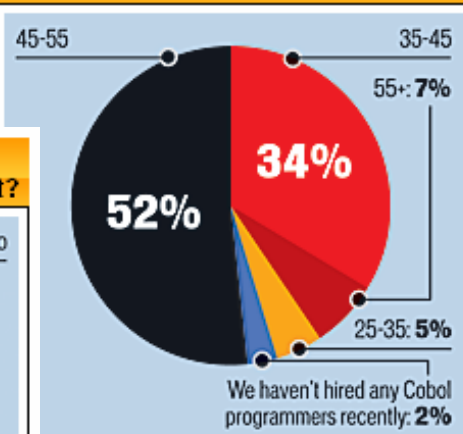
Nevertheless, with a few exceptions, companies aren't enthusiastically expanding their use of Cobol. In the survey, of those who use Cobol, 36% said they are "gradually migrating away" from it, 16% said they will replace it "every chance we get," and 25% said they'd like to replace Cobol with something else but have found that too difficult or too expensive.

In the *Computerworld* survey, 45% of the respondents whose organizations use Cobol said their ability to hire Cobol programmers was either "worse" or "much worse" than their ability to hire programmers for modern languages such as Visual Basic, C++ and Java.

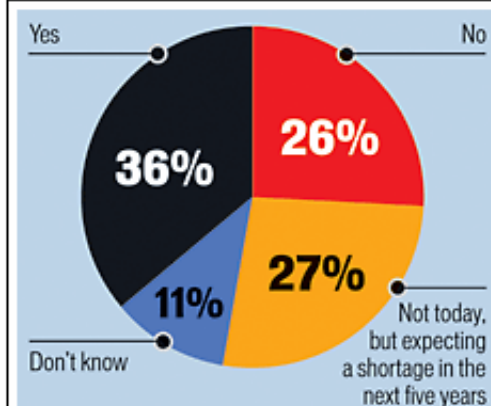
... carefully before rewriting those applications in another language. Cobol is easier to read and manage than C# or Java, says Crego, who calls Visual Basic, C and C# "write-only code." And rewriting some Cobol programs can require four or five times as many program lines in Java or C#, says Vecchio. He describes such projects as "a maintenance nightmare waiting to happen."

MORE RELATED CONTENT

What's the average age of your Cobol programmers hired in the past 12 months?



Are you noticing a shortage of Cobol programming skills in the labor market?



A simple Google search..

<http://www.networkworld.com/newsletters/nsm/2006/0828nsm2.html>

NETWORKWORLD

PARTNER SITES

- Secure Routing
- ProCurve Networking



[New! Watch this Network World Webcast on Network Management.](#)

IBM has stated that it anticipates twice as many mainframe transactions by 2010 as are running today. The expectation that this will be due in large part to service-oriented architecture (SOA) as underlined by multiple announcements throughout 2006, the most recent of which include mainframe-related [Tivoli announcements](#) and the recent [acquisition of Webify](#), a vendor of reusable SOA software components for the healthcare and insurance industries.

The legacy systems are still relevant and reliable

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

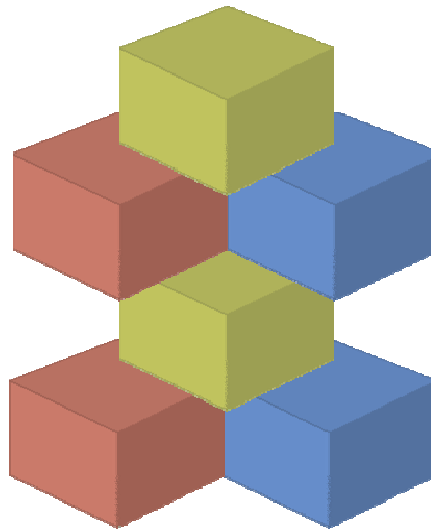


SOA: The focus is on flexibility and reuse

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

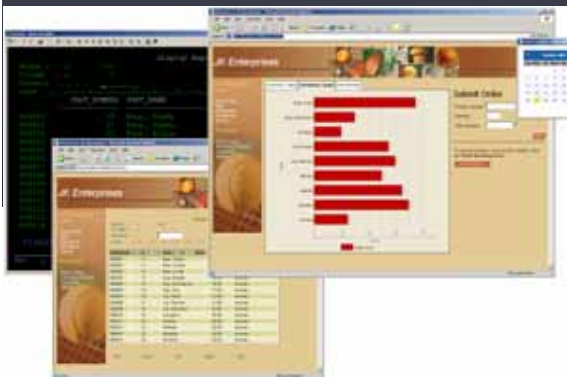
Why Service Oriented Architecture? ...

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

Three styles of application transformation

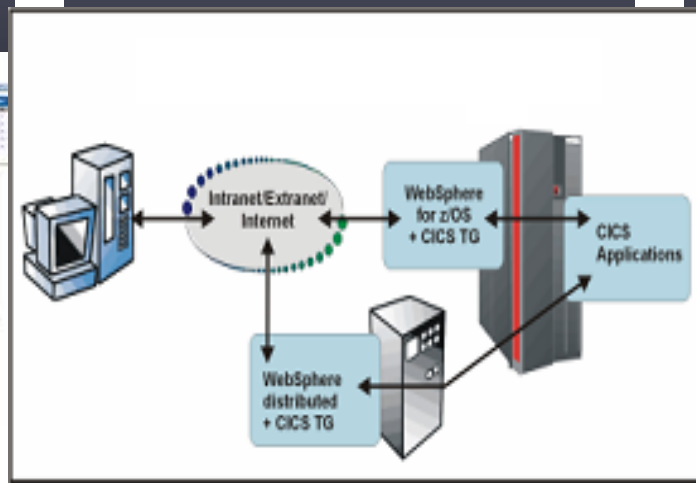
Transform User Experience

Enhance user interface and workflow for quick return on investment



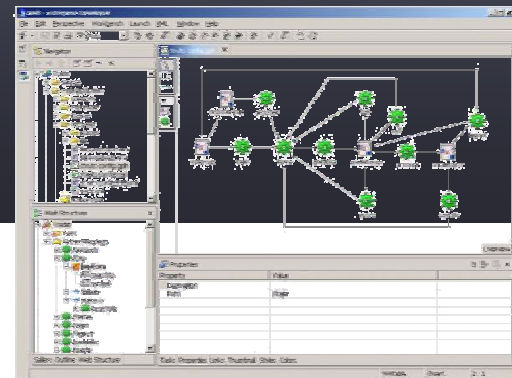
Transform Application Connectivity

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

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

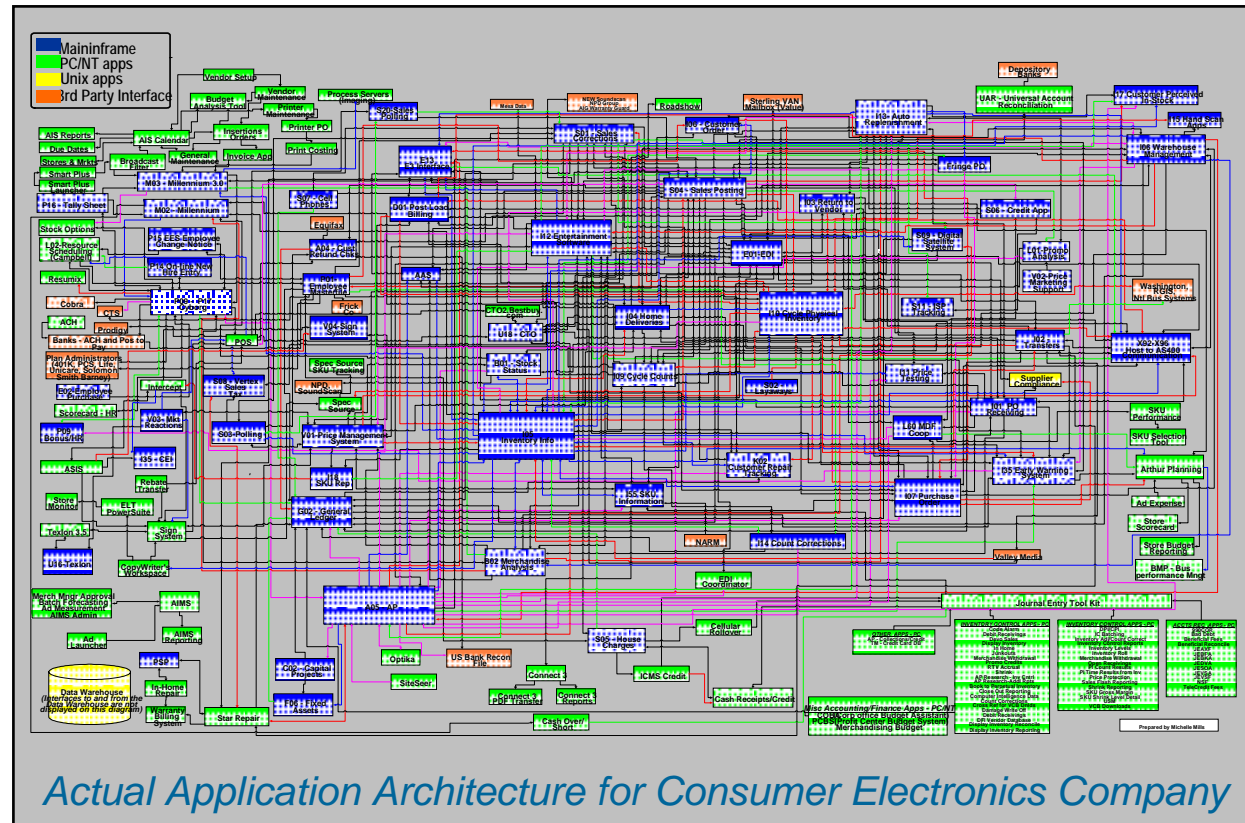


Issues: How do I?

- 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

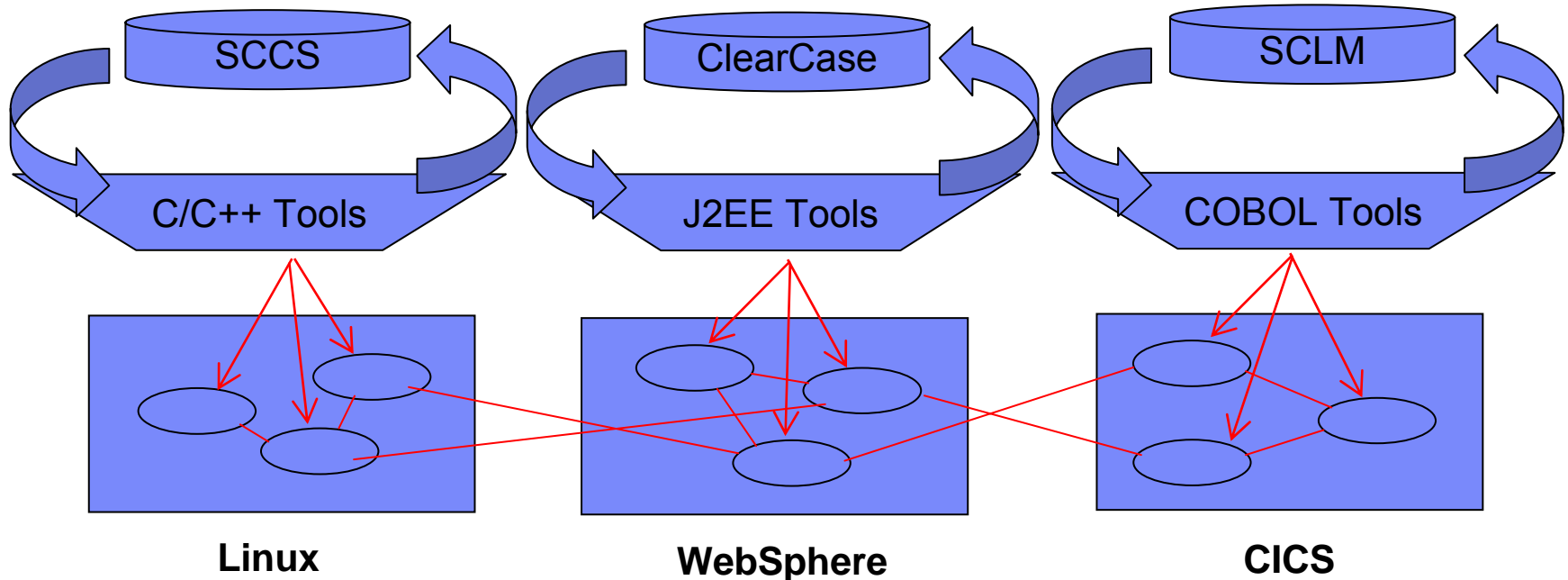


Issues: How do I?

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

Organizational challenges

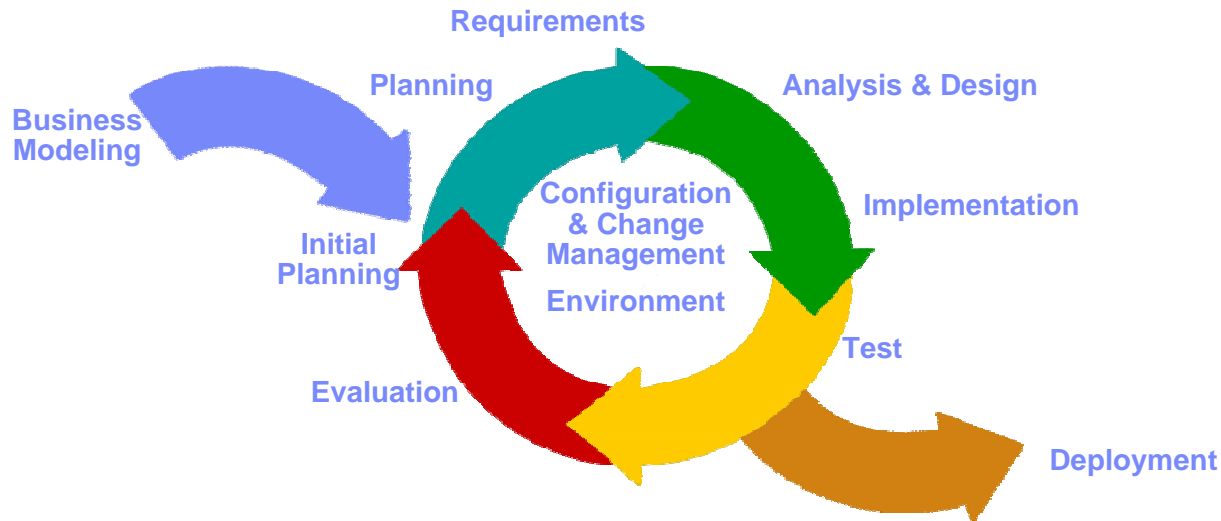
- Lack application components & skills sharing
- Ineffective / Uncoordinated development of integrated application



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



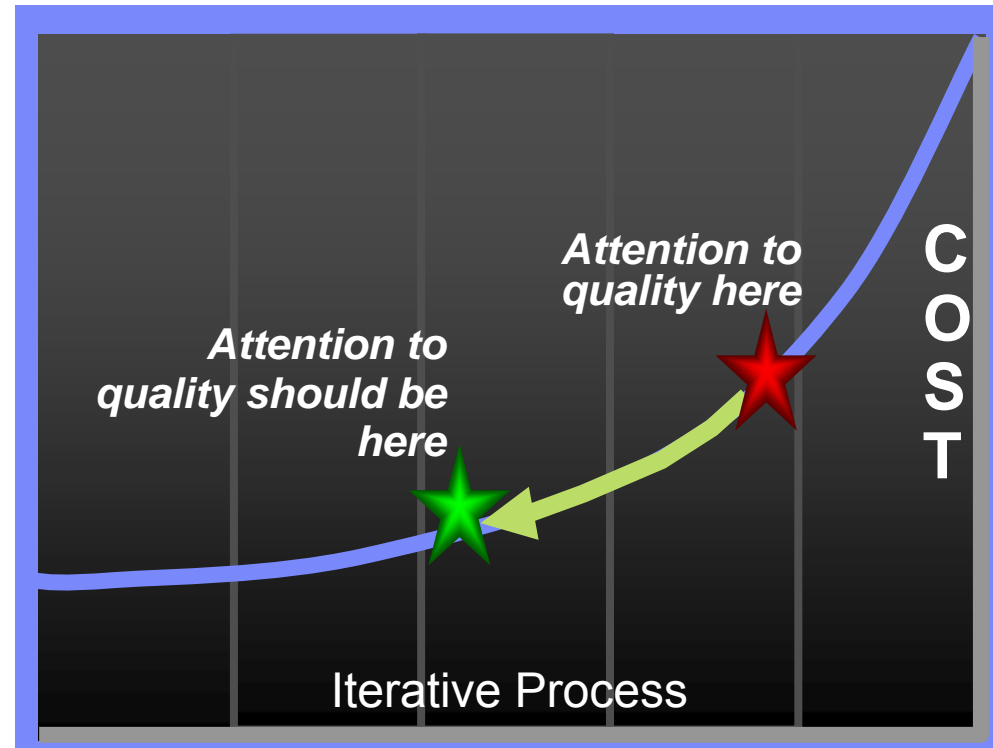
- Adopt a flexible process for both J2EE & traditional z/Series applications
- Tools integration across the lifecycle (Model and Discover, Develop and Assemble, & Deploy and Manage)
- Manage mixed workload requirements

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

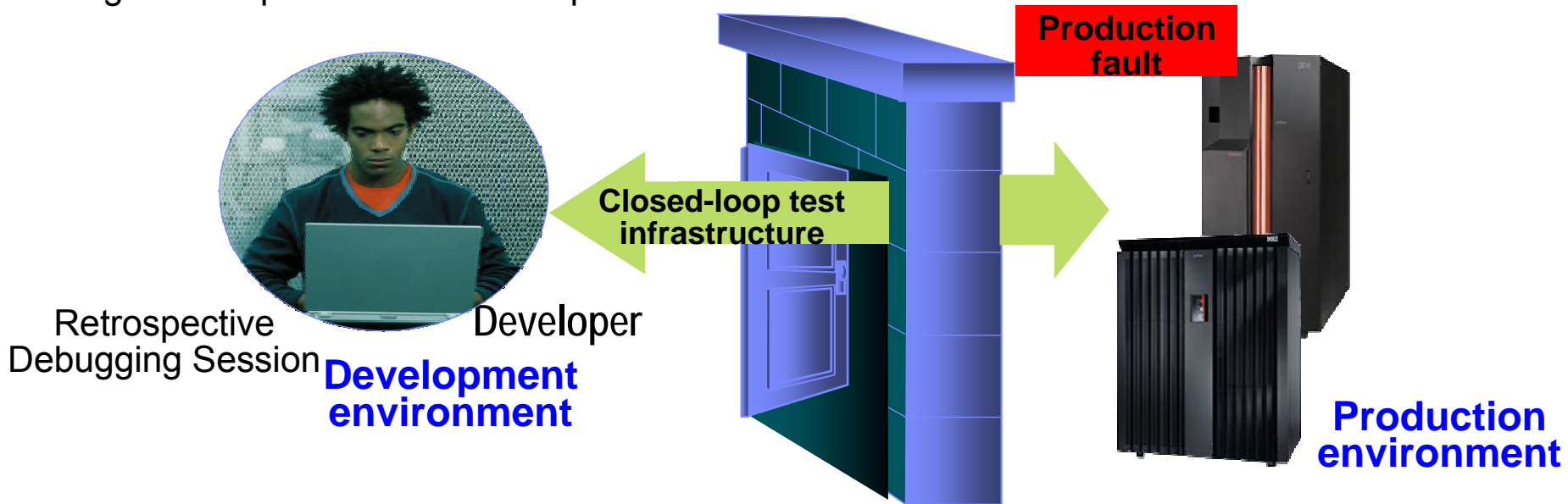


Issues: How do I?

- 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

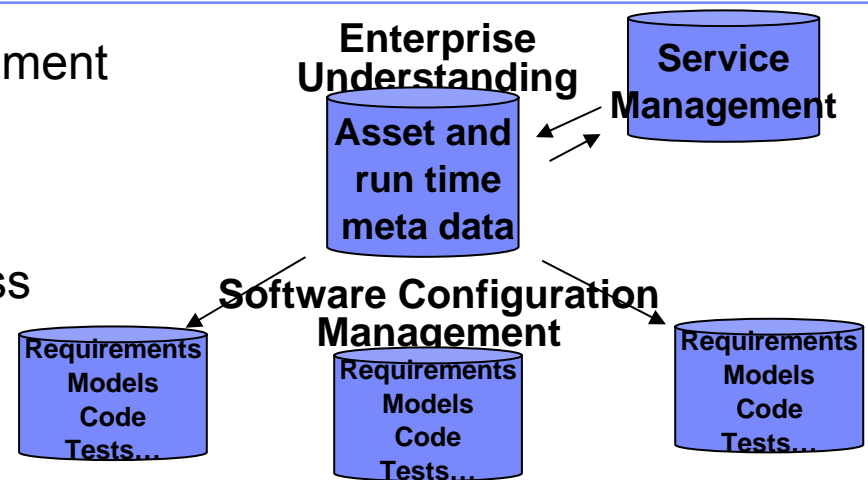


Issues: How do I?

- 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



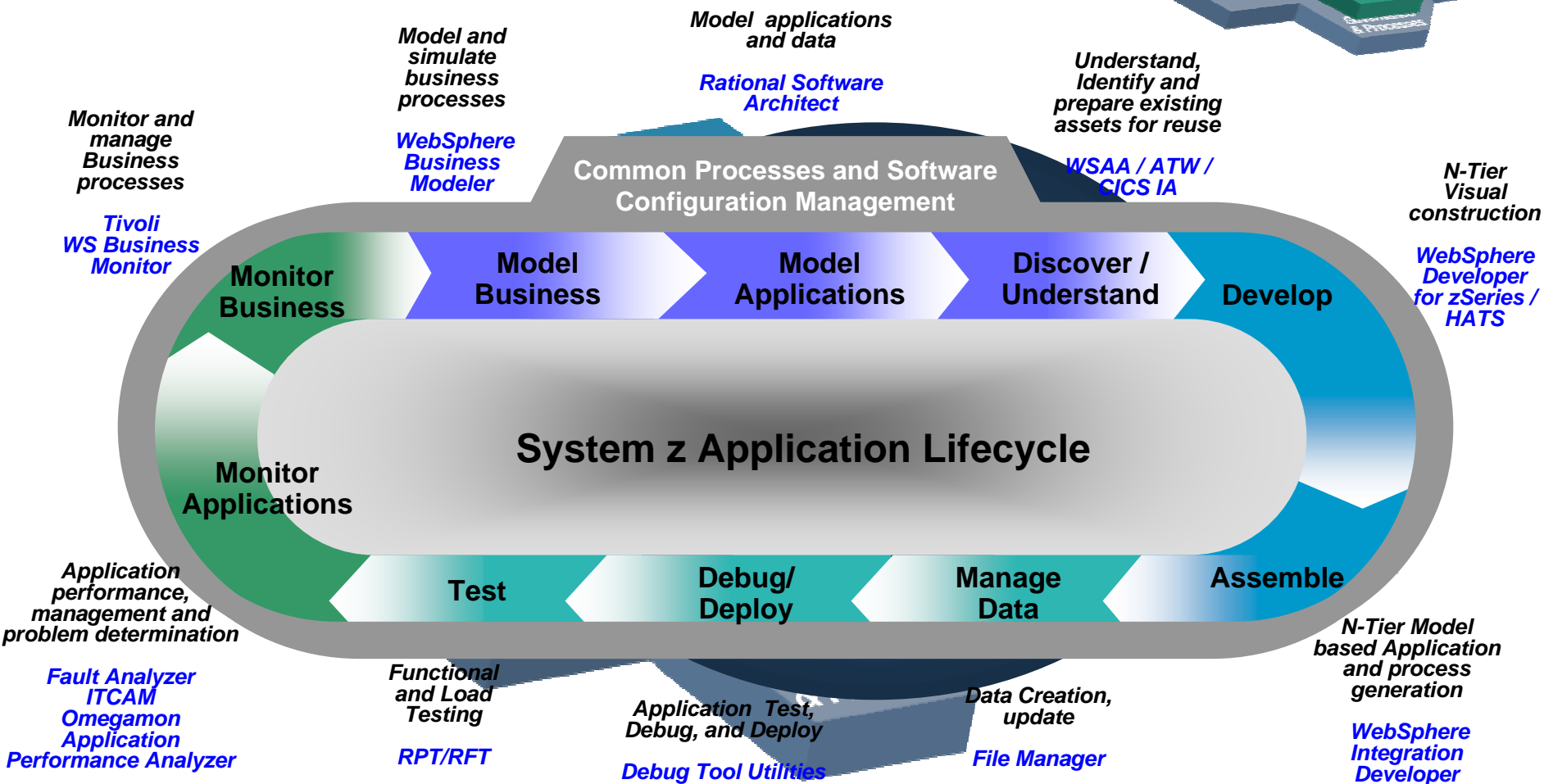
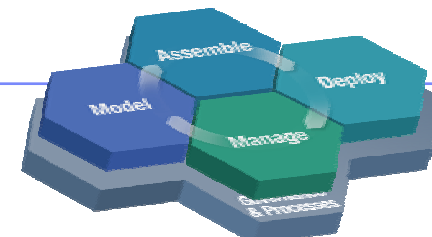
Technology Benefits

- 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

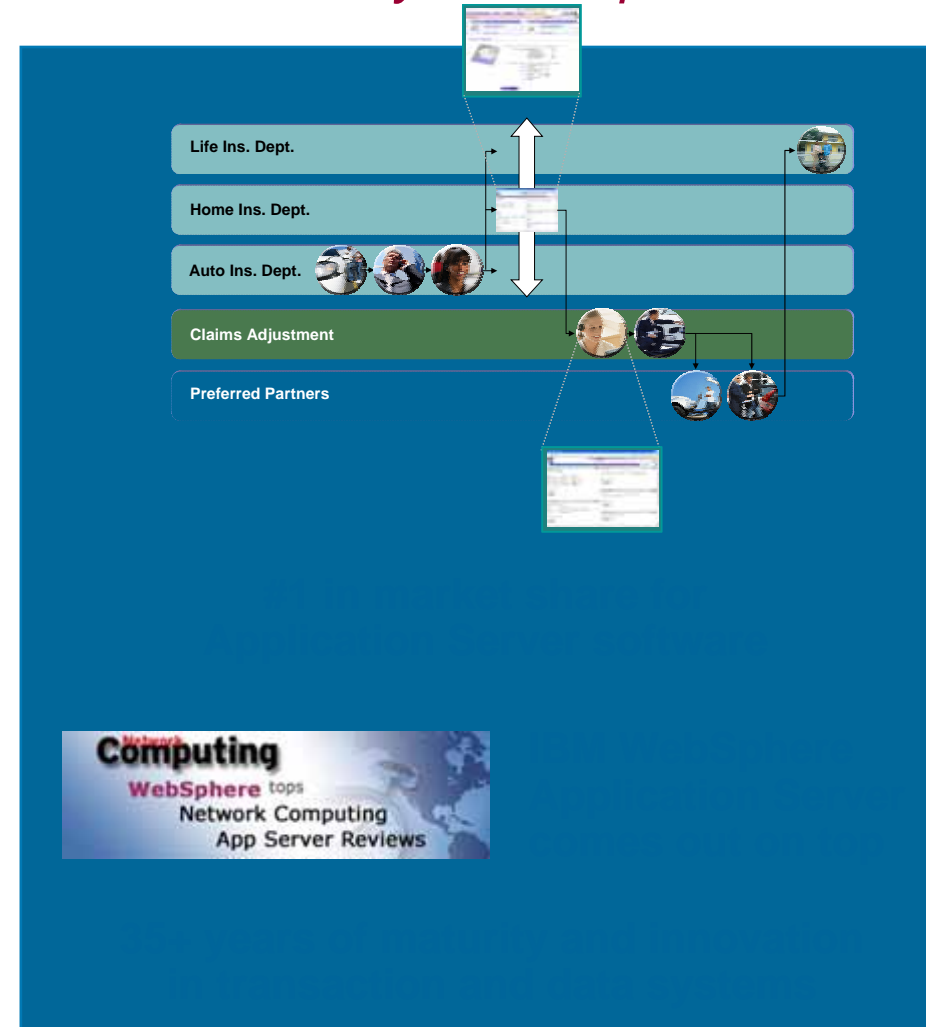
- Extend existing Java assets with support for Web Services standards and standards-based messaging
- Help ensure 24x7 availability of business-critical applications with clustering and high availability
- Build and deploy Web Services quickly and easily with rapid development and deployment features

CICS Transaction Server V3.1

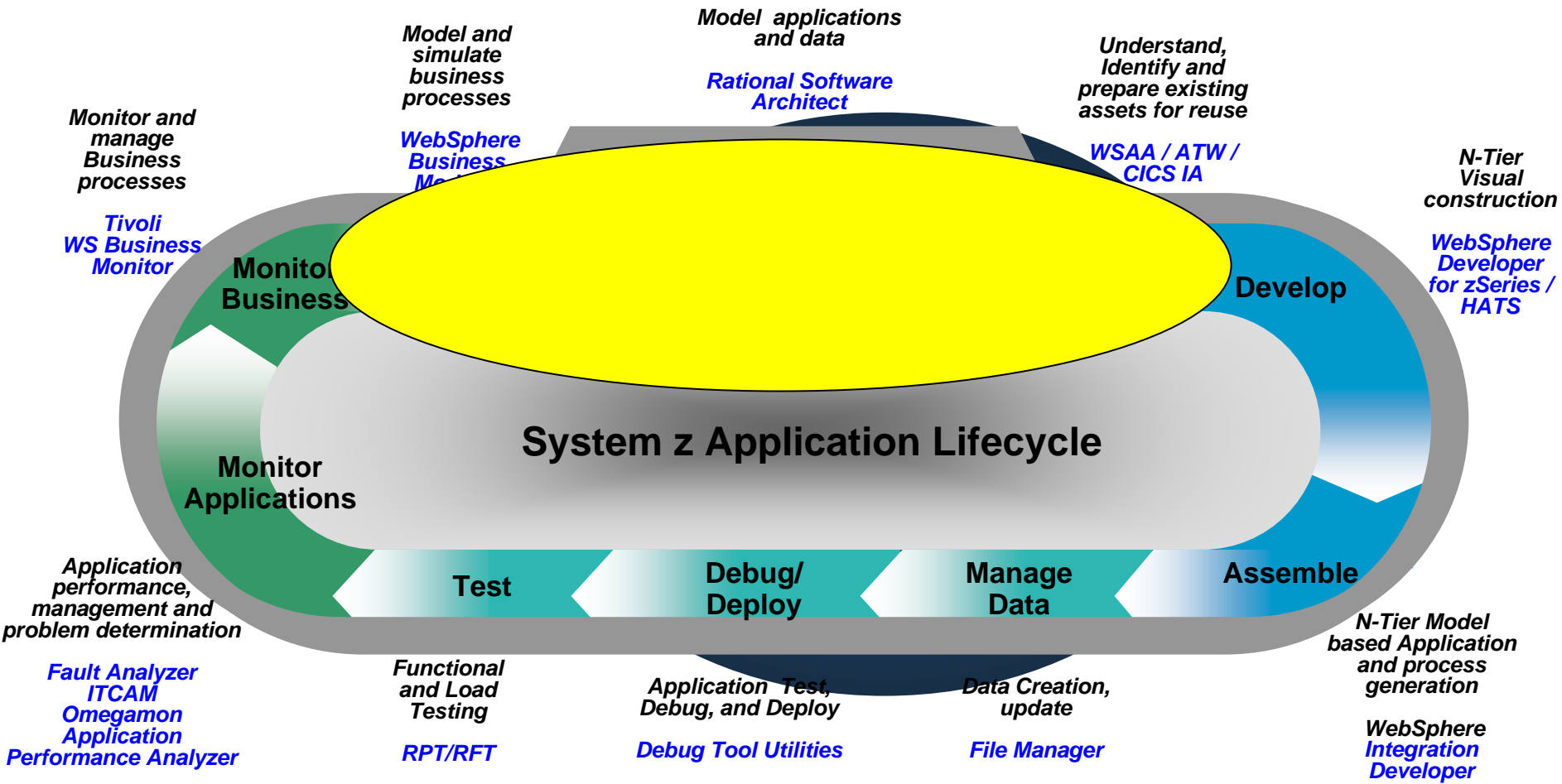
- Exploit provider/requestor Web service support for CICS assets, based on full Web service standards
- Extend the value of CICS transactions in a mixed language environment
- Build Web services from CICS transactions with no change to existing applications.

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

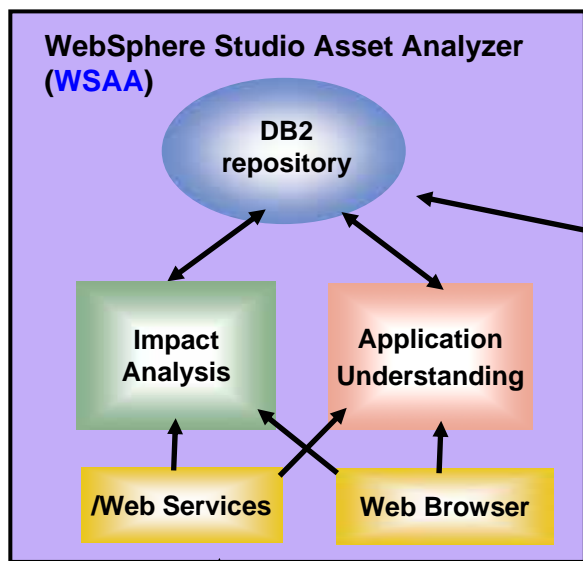


Model and discover



Discover, understand, reuse assets

Speed application discovery, understanding and asset reuse



Enterprise Customer AD artifacts

Java, COBOL, PL/1, Assembler

CICS/IMS Applications

WebSphere Applications

DB2, WSMQ

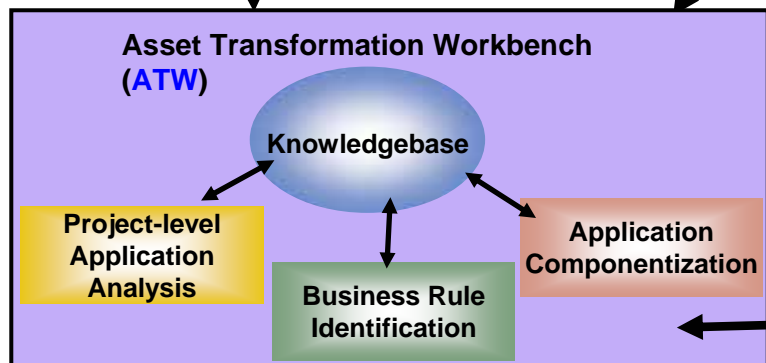
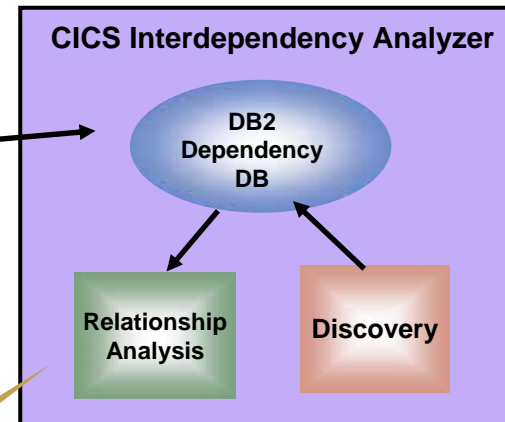
CICS Application Resources

Transactions

Programs, Files

TDQs, TSQs

DB2/IMS DB, etc.



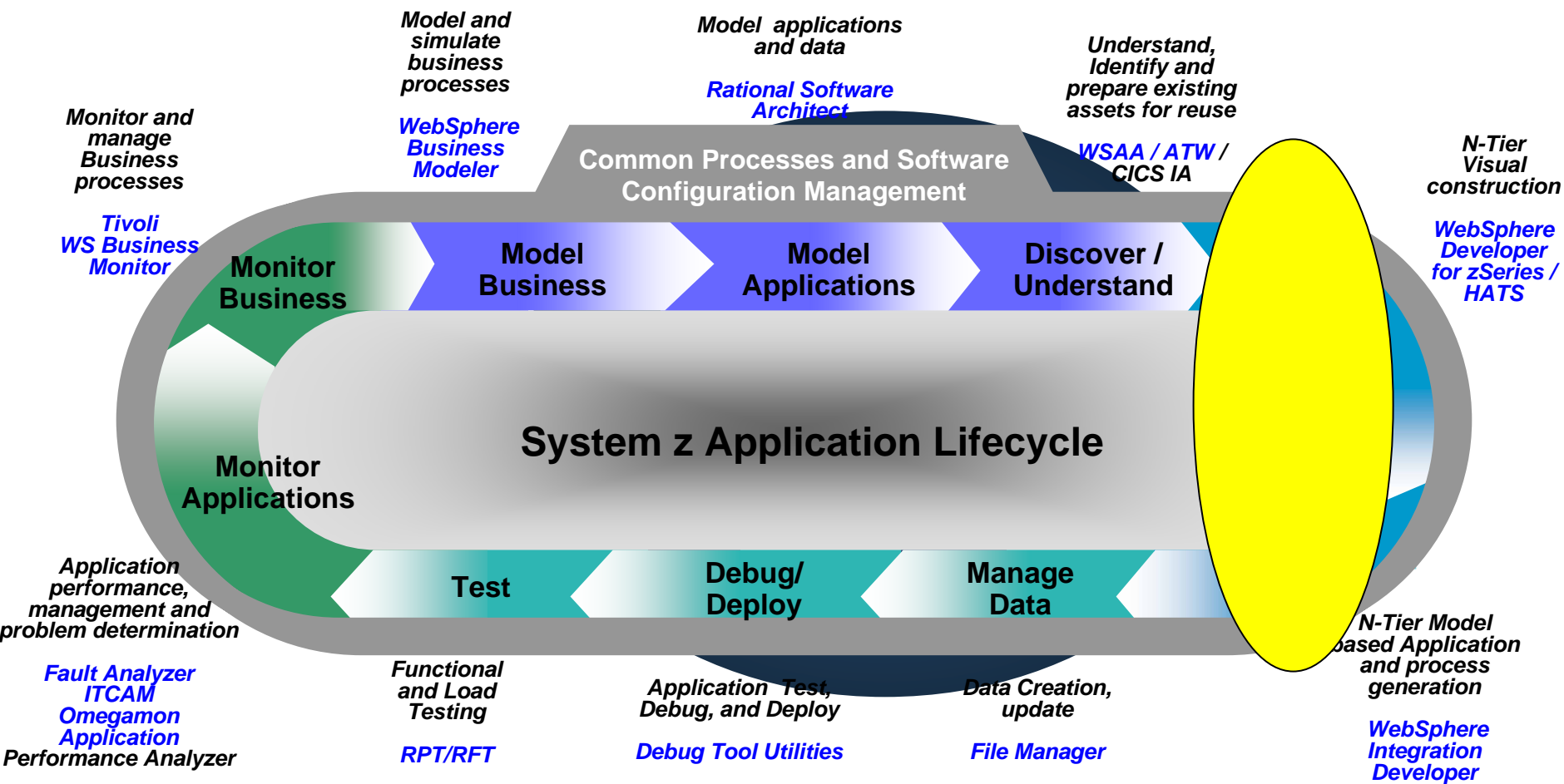
Users: business analysts, system analysts, developers, testers, project managers, management, System Programmers, QA analysts

Architects, project leaders

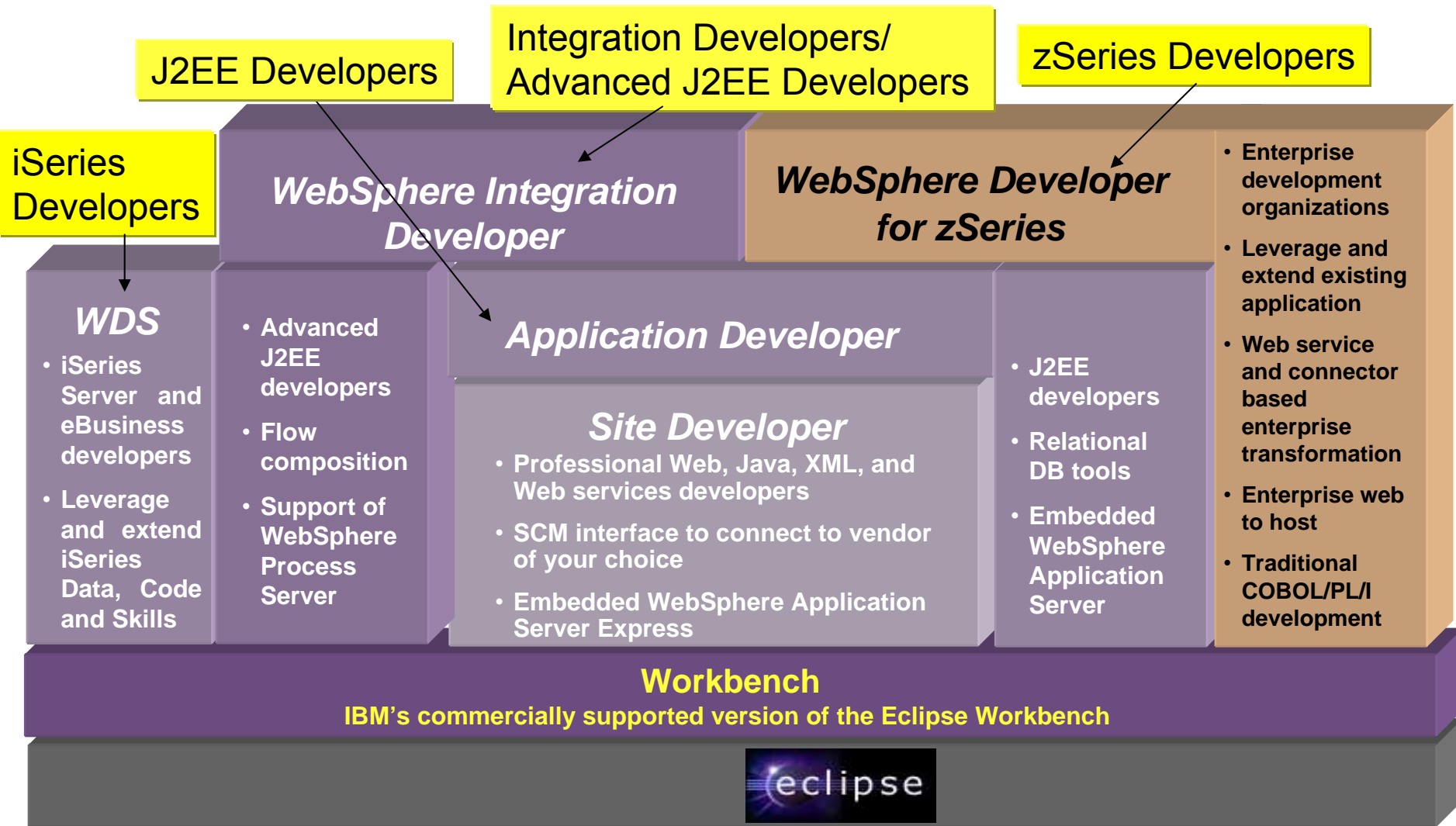
Benefits:

- Automated discovery of application code and CICS runtime relationships
- Higher quality of application change management
- Reduce or eliminate intensive efforts to create components
- Position for evolution SOA

Develop and assemble



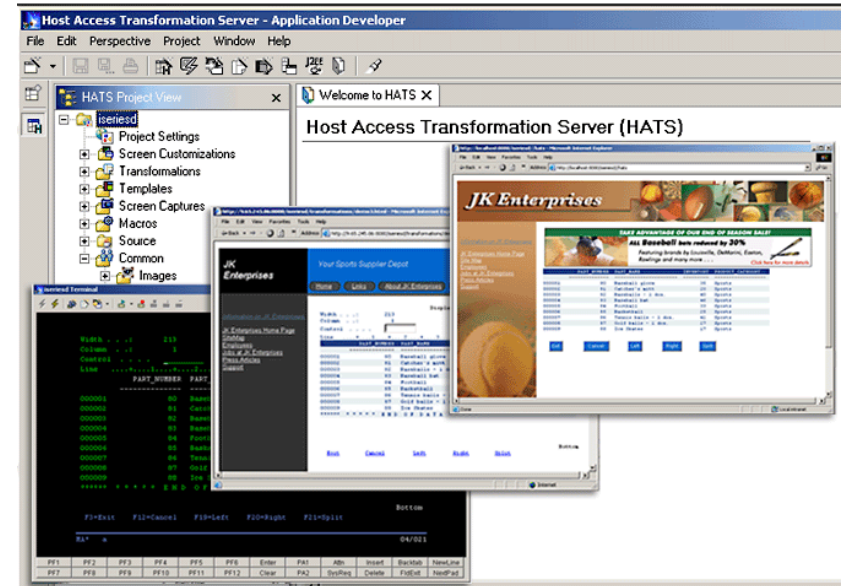
WebSphere/Rational development family



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 zSeries skills required
- Rules-based, highly customizable
- Interactive, eclipse-based development environment

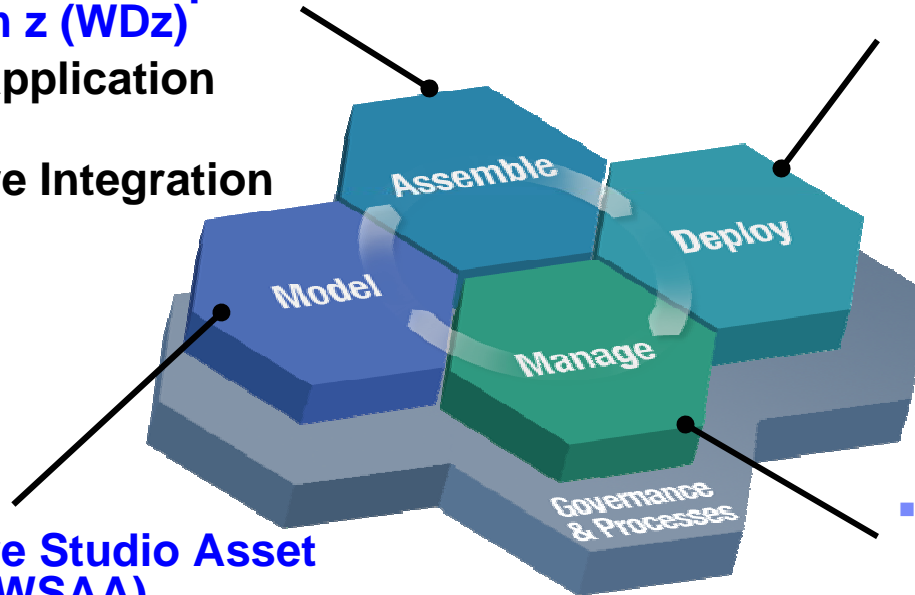


Benefits:

- 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.

SOA: Reuse and service generation

- **WebSphere Host Access Transformation (HATS)**
- **WebSphere Developer for System z (WDz)**
- Rational Application Developer
- **WebSphere Integration Developer**



- **WebSphere Studio Asset Analyzer (WSAA)**
- **Asset Transformation Workbench (ATW)**
- **WebSphere Business Modeler**

- **WebSphere Application Server**
- **WebSphere Portal Server**
- **WebSphere Process Server**
- **CICS**
- **IMS**
- **TPF**

- **Tivoli Composite Application Manager**
- **Tivoli for SOA**
- **WebSphere Registry and Repository**
- **WebSphere Business Monitor**

In blue are the products covered in this session

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.
- Reasons that COBOL continues 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).