



# The New Face of Application Development for System z

*A. Hayden Lindsey  
IBM Distinguished Engineer  
Director, Rational System z and System i*



2006 System z Premier Event

## Agenda

- Key messages
- Today's realities
- Reshaping software development
- IBM's software strategy for System z and multi-platform



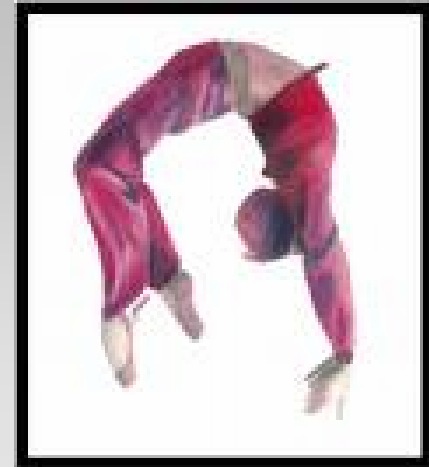
## Agenda

- Key messages
- Today's realities
- Reshaping software development
- IBM's software strategy for System z and multi-platform



## Key Messages

- ✓ Businesses need to change to stay viable, and IT must enable it
  
- ✓ To be sufficiently nimble, there are several enablers that we are exploiting
  - Community & Open Computing
  - Modularity
  - Governance
  
- ✓ The IBM Software Development Platform provides world-class application development support for System z
  - Development & deployment of SOA solutions is easy and efficient
  - IBM provides the integrated tools, team infrastructure and governance platform to help your existing and future staff productivity create new solutions and also maintain the existing applications that run your business



# Agenda

- Key messages
- Today's realities
- Reshaping software development
- IBM's software strategy for System z and multi-platform



# Enterprise pressures and opportunities

commoditization pressures

new/increased competition

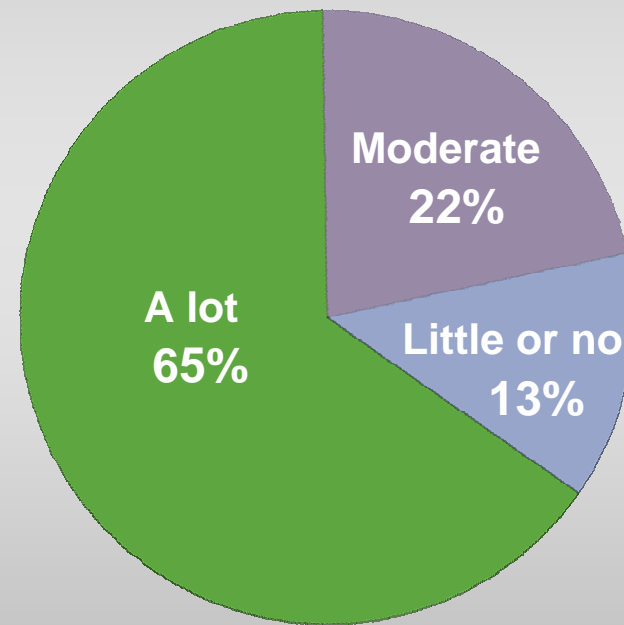
global market opportunities

adjacent market opportunities

global volatility & disruption

competing business models

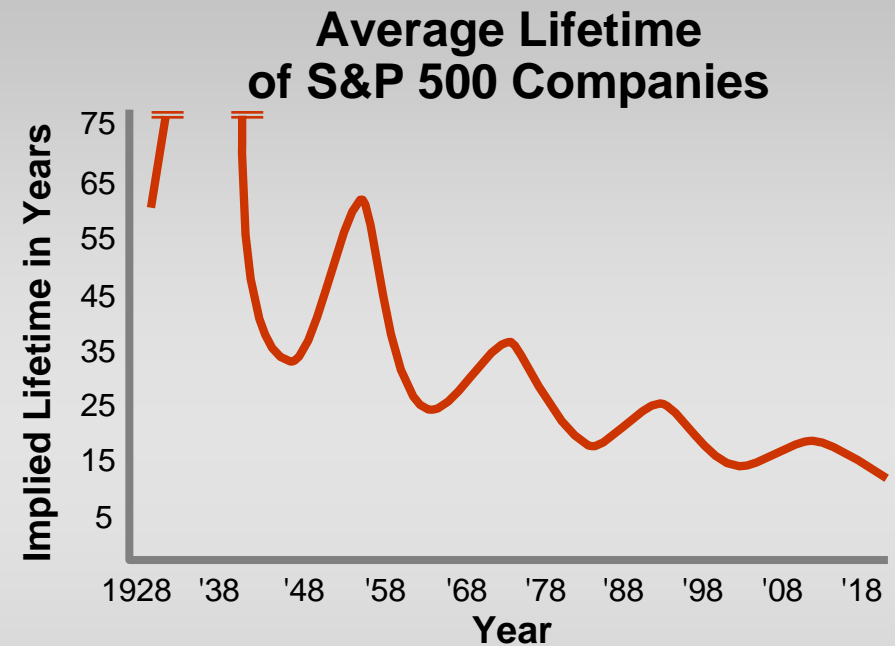
CEOs: Extent of fundamental **change needed** over next two years



IBM Global CEO Study 2006

## Marketplace destabilization - *and it's accelerating*

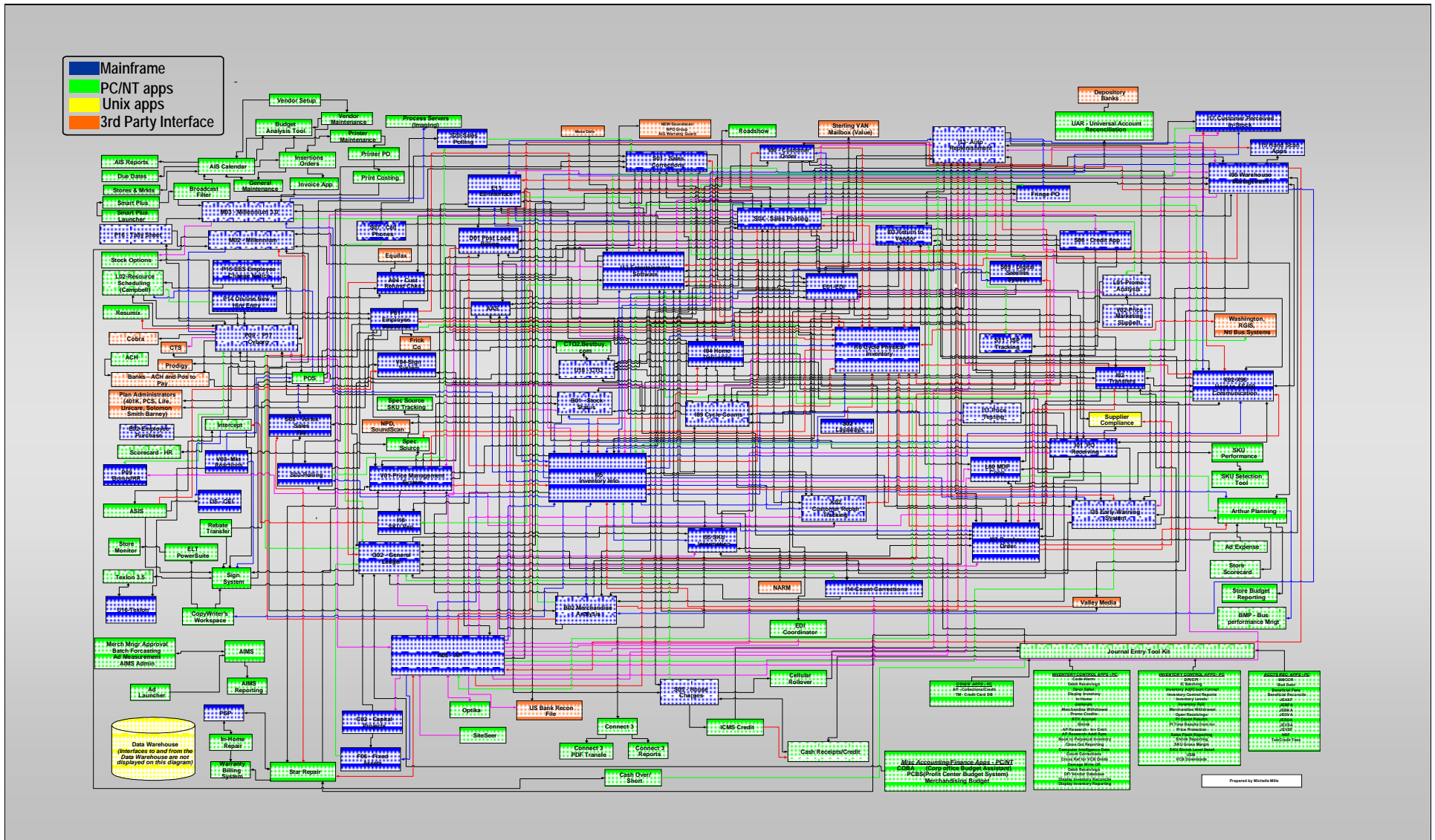
- ▶ *Technology* systematically reduces interaction costs and extends global reach
- ▶ *Globalization* increases complexity of business requirements and IT agility
- ▶ Constant global policy shifts alter *regulatory* and competitive climates
- ▶ Intense pressure on *business models* drives focus on core competencies



Source: *Creative Destruction*, by Richard Foster

*Destabilizing forces converge  
to significantly intensify  
global competition*

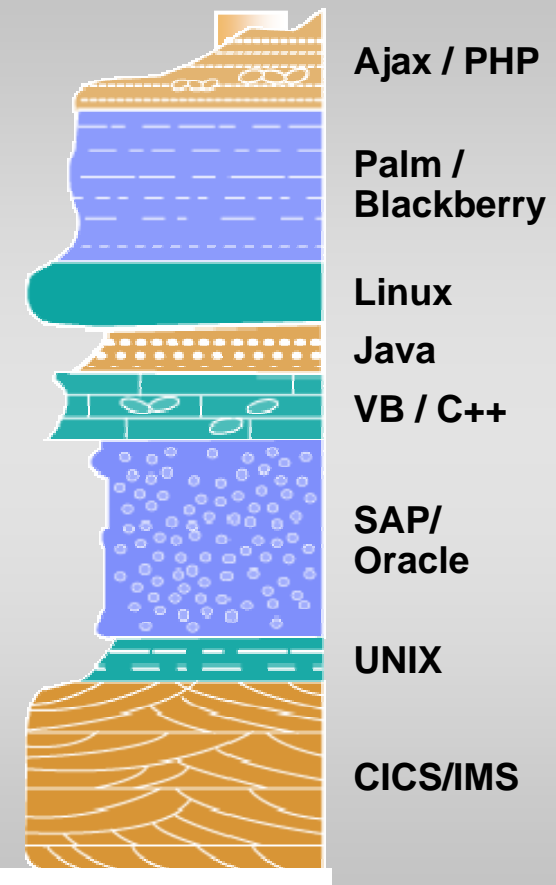
# Software engineering realities – complex, tightly coupled architectures





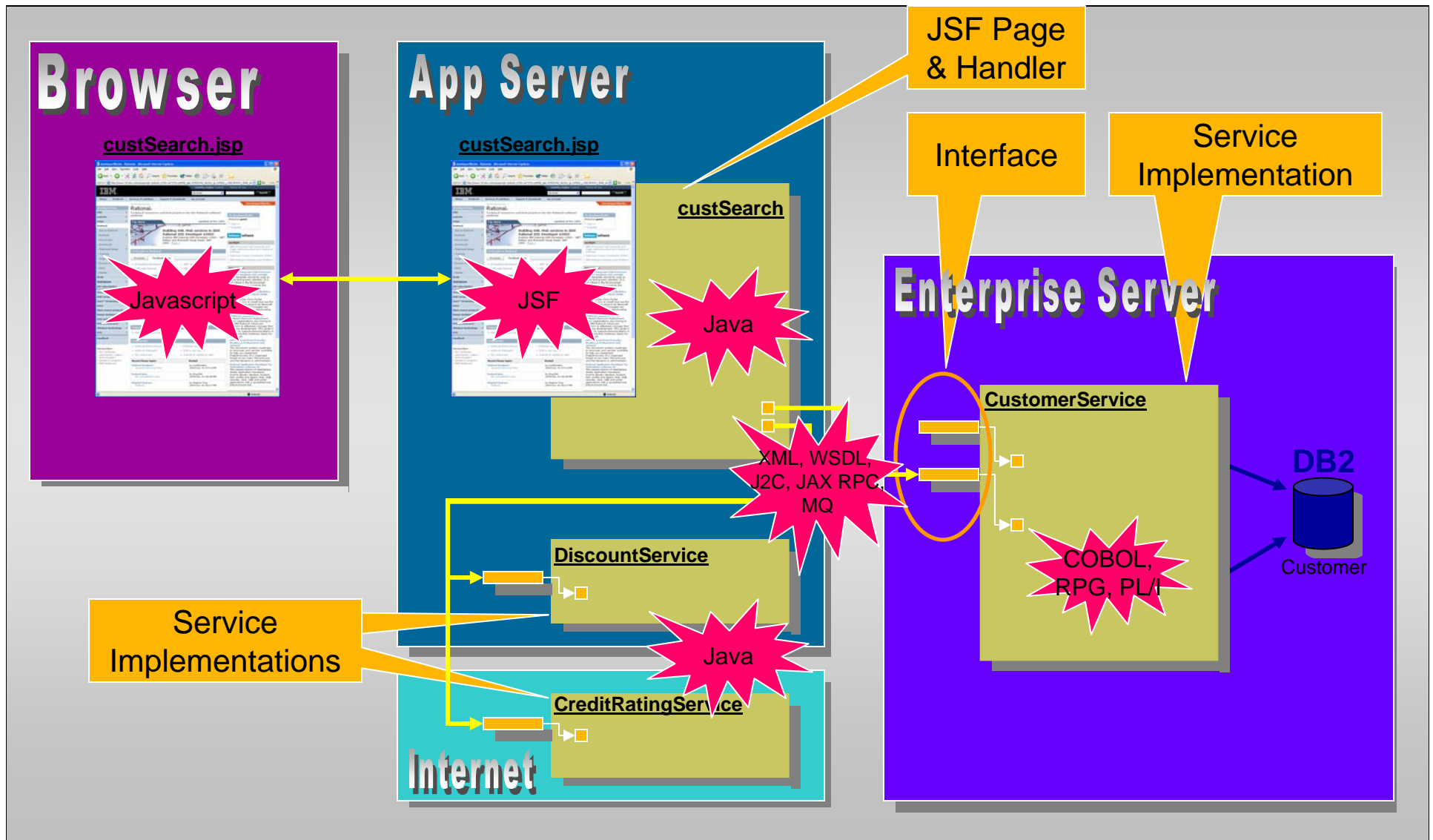
## Software engineering realities – *large variety of middleware*

- ▶ In contrast to physical computing
  - Software evolution is constrained by decades of legacy code
  - Agility is constrained by layers
  - Value comes in automation of new business abstractions, rules and models
- ▶ Chaos results from
  - Multiple generations of ‘captured intelligence’ in the form of code / business rules
  - Mixed with new generations of technology assumptions (mainframe to C/S to peer distributed – and variants)
- ▶ Software archeology or software architecture?



Source: “*The Agile Dance of Architectures*”, by John Hagel, III and John Seely Brown

# Software engineering realities - *many technologies; who has the skills?*



Today's realities

Software Development  
Process, Discipline and  
Productivity must accelerate

# Today's realities

## Accelerators

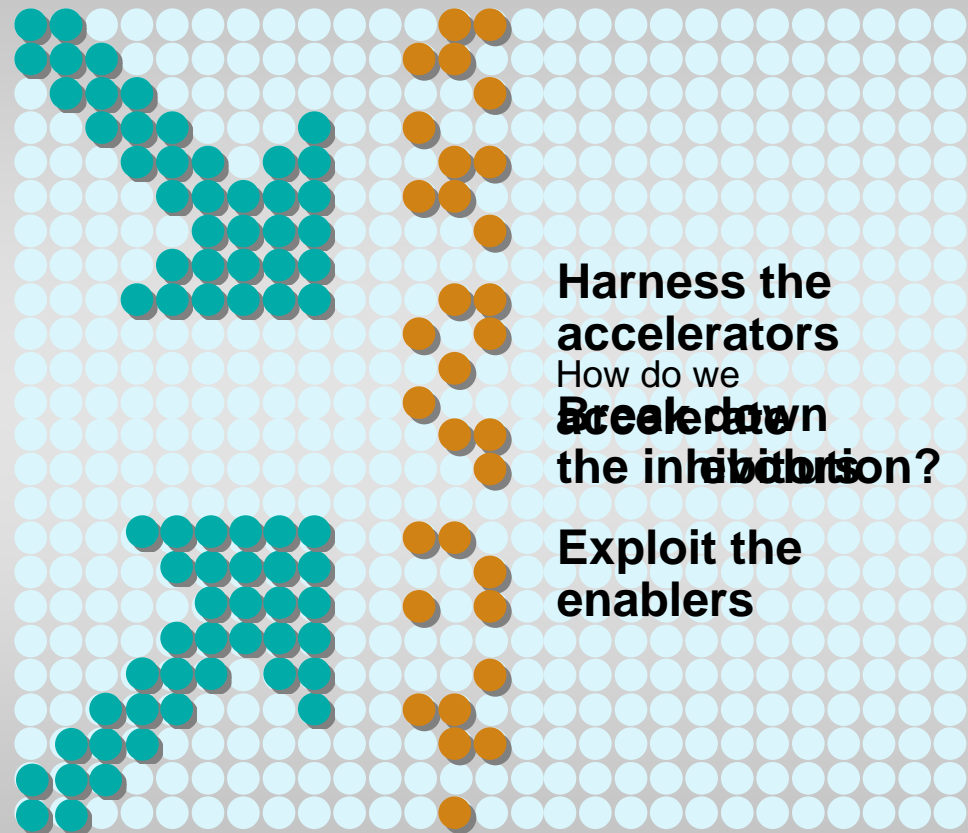
- Intense global competition
- Expanding regulatory requirements

## Inhibitors

- Complex, tightly coupled architectures
- “Sedimentary Layers” of middleware stacks / systems
- Culture, processes and skills of development teams

## Enablers

- Moore’s Law drives physical computing limits
- Bandwidth capacity far exceeds demand



## Agenda

- Key messages
- Today's realities
- Reshaping software development
- IBM's software strategy for System z and multi-platform



# Reshaping software development

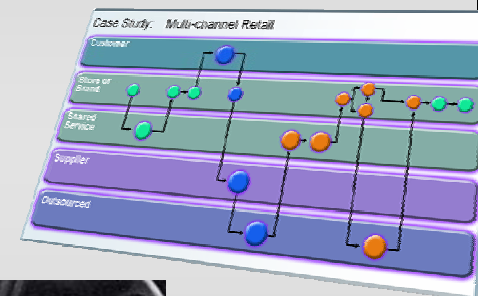
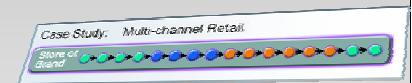
- ▶ Open Computing & Communities
  - Integrate more often and more easily
  - Leverage community effects from open computing, Metcalf's law, social networking
  
- ▶ Modularity
  - Exploit SOA as a key enabler of business flexibility
  - Do so on all platforms – it is an architecture!
  
- ▶ Empowerment and innovation via passive governance
  - Maximize value and flexibility of the knowledge-based workforce
  - Minimize chaos while maximizing individual decision rights

Open  
architecture

Open  
computing

Open  
standards

Open  
source



## Open computing - a new route to collaboration, innovation, integration

- **Open standards**
  - ▶ Improve data sharing and runtime interoperability
- **Open architecture**
  - ▶ Increase business flexibility via loose coupling, component-based architectures
- **Open source**
  - ▶ Promote innovation, quality and timeliness by leveraging community development

Open  
architecture

Open  
computing

Open  
standards

Open  
source

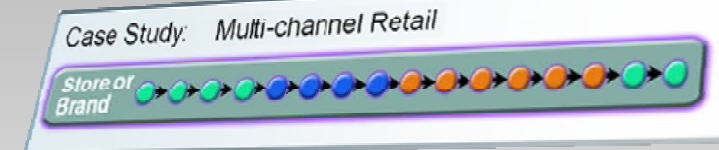
# Modularity

SOA is a key enabler for Business Flexibility – the latter is the goal, so IT must understand the former

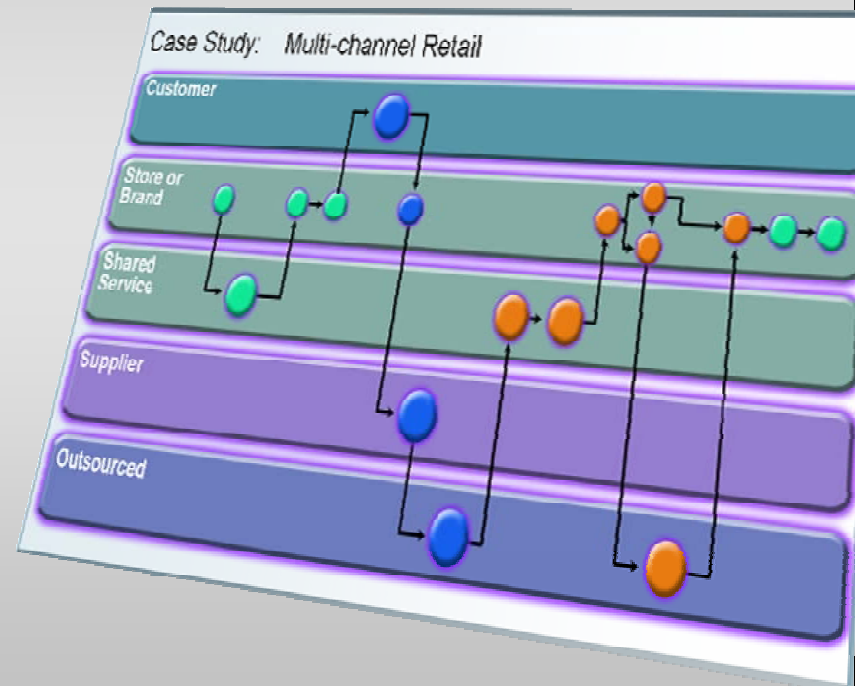
- **Competitiveness:** rate of change demands IT flexibility
- **Growth:** at the top of the CEO agenda
- **Economics:** reuse can cut costs

The picture to the right is “logical”. This could all be running under CICS.  
 “Flexibility” is the key point.

## Traditional Business\*



## Today's World-Class Business\*



\*Sources:  
 CBDi



## Modularity – *considerations for software development*

- IBM is
  - ▶ Providing cross-lifecycle tools that are “service-aware”
  - ▶ Providing runtimes that improve & abstract Service support
  - ▶ Providing these capabilities in a consumable, incremental fashion (not all-or-nothing)
    - Modeling (business & application), language (EGL), visual wiring, testing, Web Services, WPS, monitoring, etc.
  - ▶ Providing best practices & governance support to increase chances of success

## Empowerment through Governance

- ▶ We have learned lots from our participation in Apache and Eclipse
  - Our history was extreme “cathedral”-style development\*
  - Moving to an open-source style was a difficult & large cultural & practical challenge
  - Cost of building community was outweighed by benefits
  - Improvement in quality and predictability was significant
- ▶ Top-down imposed governance fails unless developers benefit too
- ▶ Process, governance, and auditing need to be part of the day-to-day activity, not “extra work”
- ▶ The keys:
  - Integrated, flexible process
  - Automation
  - Visible, timely information that supports decision making



\*Source: “*The Cathedral and the Bazaar*” by Eric S. Raymond

## Agenda

- Key messages
- Today's realities
- Reshaping software development
- **IBM's software strategy for System z and multi-platform**



# What we hear ...

**Increasing Business Value**

**Ford**

**FIDELITY NATIONAL FINANCIAL**

We want you to know™  
**Aetna**

**Danske Bank**

**"la Caixa"**

**Fifth Third Bank**

**TATA**

**Nationwide**  
On Your Side™

**delta lloyd groep**

**STATE FARM INSURANCE**

*How do I better manage and **govern** the development and integration of all my assets regardless of platform?*

*How do I utilize my existing development **talent** and **skills** to build new applications that leverage modern service oriented architectures?*

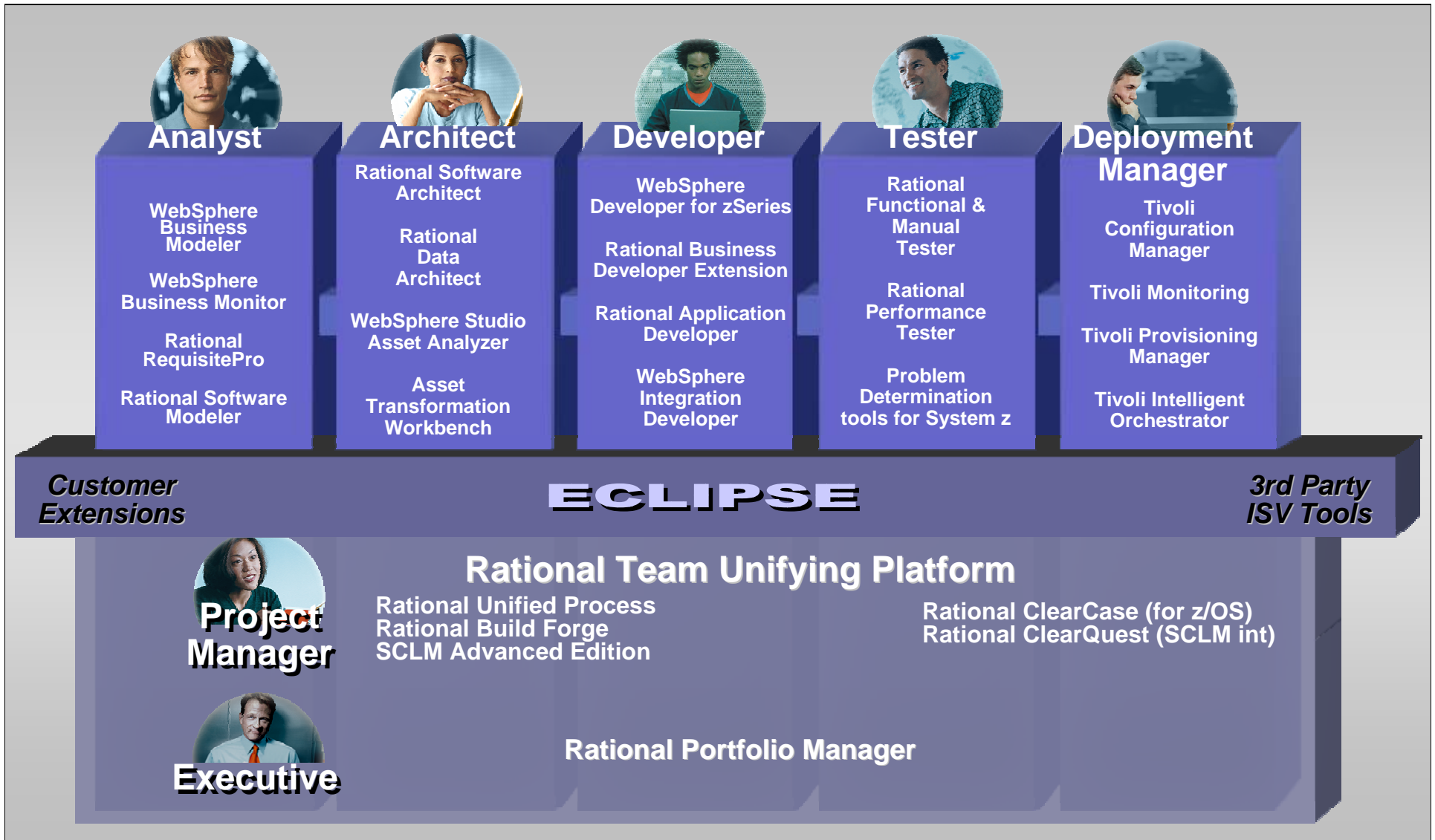
*How do I **preserve my investment** in my System z assets by making them more available in new ways for new value?*

*How can I drive down costs by making sure my developers are **productive** and delivering **high quality** applications in a **predictable** manner (on time and on budget)?*



# The IBM Software Development Platform

*Integrated and role-specific tools for SOA, z and multi-platform*



# The IBM Software Development Platform - Enterprise Style

*Enable predictable, integrated, multi-platform software delivery*

## Extending the IBM Software Development Platform to System z



- Improve developer productivity & reduce costs
  - ▶ Common processes & tools regardless of deployment platform provide greater team flexibility, productivity
  - ▶ Fewer tools means lower support & training costs
  - ▶ New tools that create web services from existing applications offers new business value
- Enhance quality & flexibility of your solutions
  - ▶ Tools to facilitate application discovery, understanding and re-factoring extract value from existing code
  - ▶ Model-driven development & SOA tools exploit latest in productivity, quality and flexible architectures
  - ▶ Best practices and tool advisors help you “do it right”
- Effectively govern enterprise development
  - ▶ Dashboards for identifying and managing project risk, monitoring and managing runtimes aid decision-making
  - ▶ Converged source code libraries & change mgmt facilitate end-to-end solution development

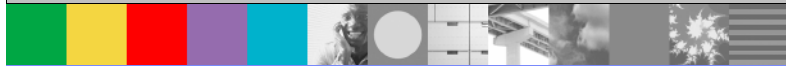
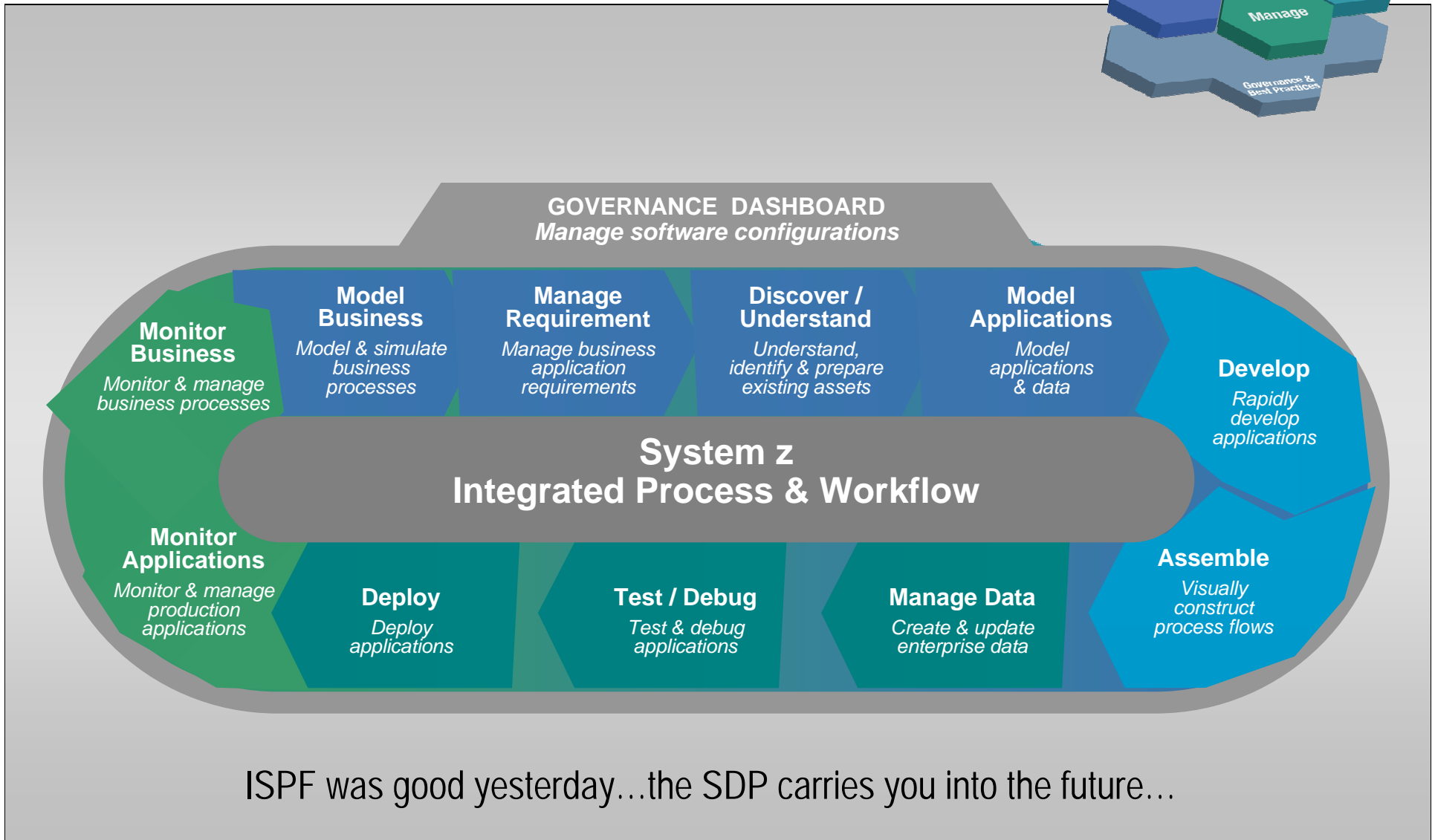
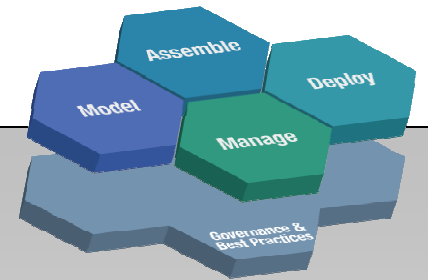




## Software Development Strategy for System z – *a few details*

- Different plans for different parts of the SDP
  - ▶ Practitioner IDE
  - ▶ Tools that create (directly or indirectly) runtime artifacts (e.g. UML x-forms, EGL)
  - ▶ Tools that use runtime artifacts (e.g. RFT, RPT)
  - ▶ Tools with data stores (e.g. CQ, RPM, ReqPro)
  - ▶ Tools with server components (e.g. CC)
  
- General strategy
  - ▶ Practitioner tools run off-platform (Win, Linux)
  - ▶ Practitioner tools support discovery, creation, deployment to System z
  - ▶ Tools that use runtime artifacts support applications on System z
  - ▶ Tools with data stores support DB2 for z/OS
  - ▶ Tools with server components run on System z
    - ...whether z/OS, USS, z/Linux depends upon cost & customer reqs

# Software Development Lifecycle





# Manage Requirements

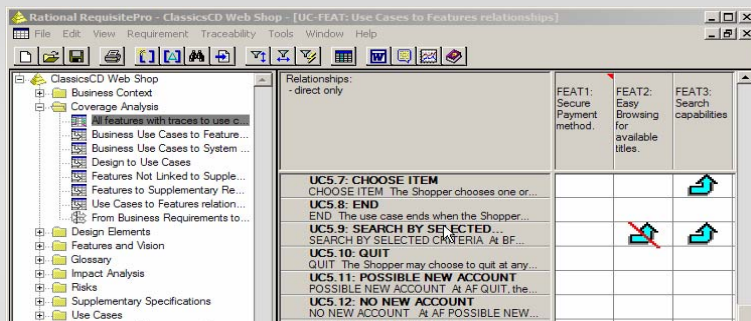
*Understand what needs doing, and ensure it gets done*

## Scenario:

- We need to formalize requirements management and trace requirements to designs, code, testcases, build records and deployment plans.

## What's here:

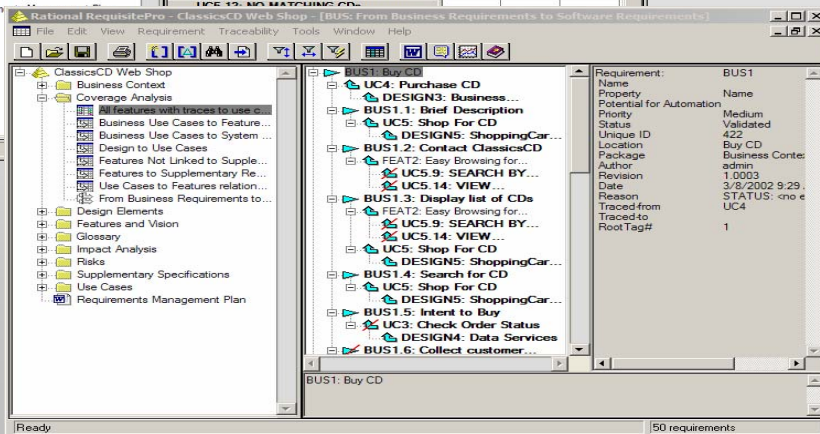
- Rational RequisitePro – handle requirements management for project teams



✓ Graphical trace matrix

## What's coming:

- Rational RequisitePro – use DB2 for z/OS



✓ Graphical trace tree

ReqPro just works, regardless of whether development is for System z, System i or distributed. Same for other parts of the SDP: RPM, WBM, RMT, etc.

# Model and Simulate Business Processes

*Model process changes and simulate savings before committing resources*

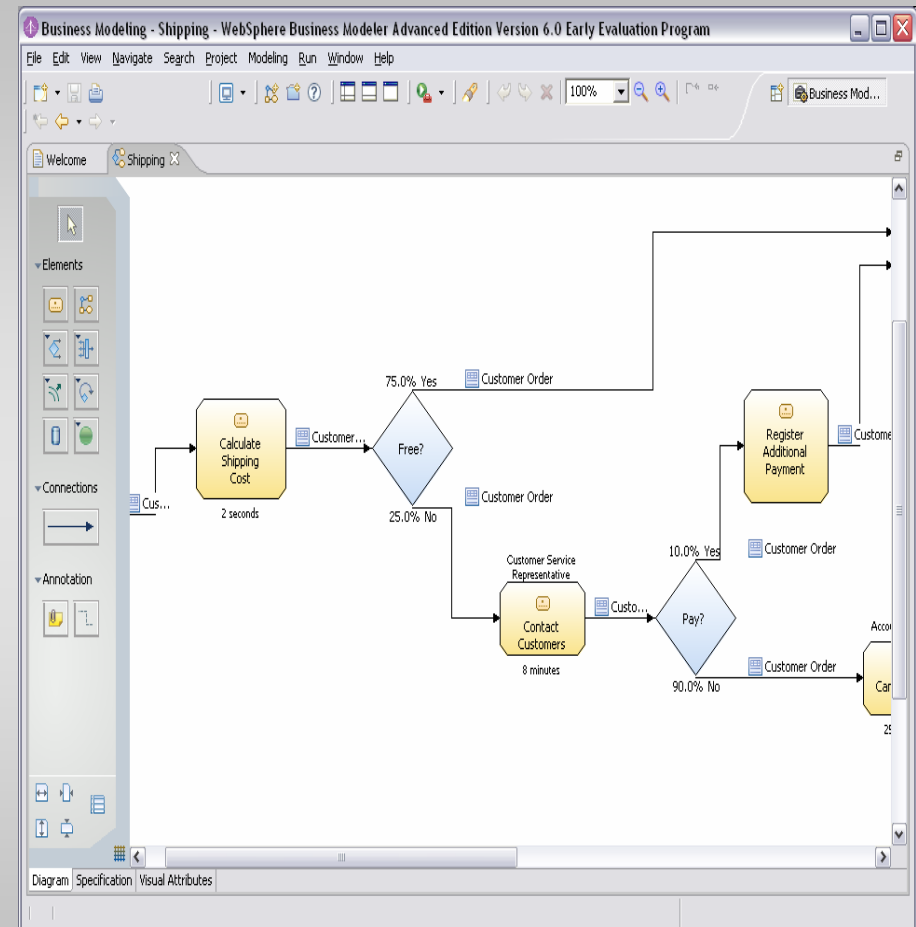
## ► Scenario:

- We need to visualize our critical business processes, and use modeling, simulation, and analysis to evaluate potential process enhancements prior to implementing changes.

## ► What's here:

- WebSphere Business Modeler - visualize, comprehend, document and improve your business process events. Implementations can be deployed to WPS on z/OS.

## WebSphere Business Modeler



# Discover, Understand & Reuse Assets

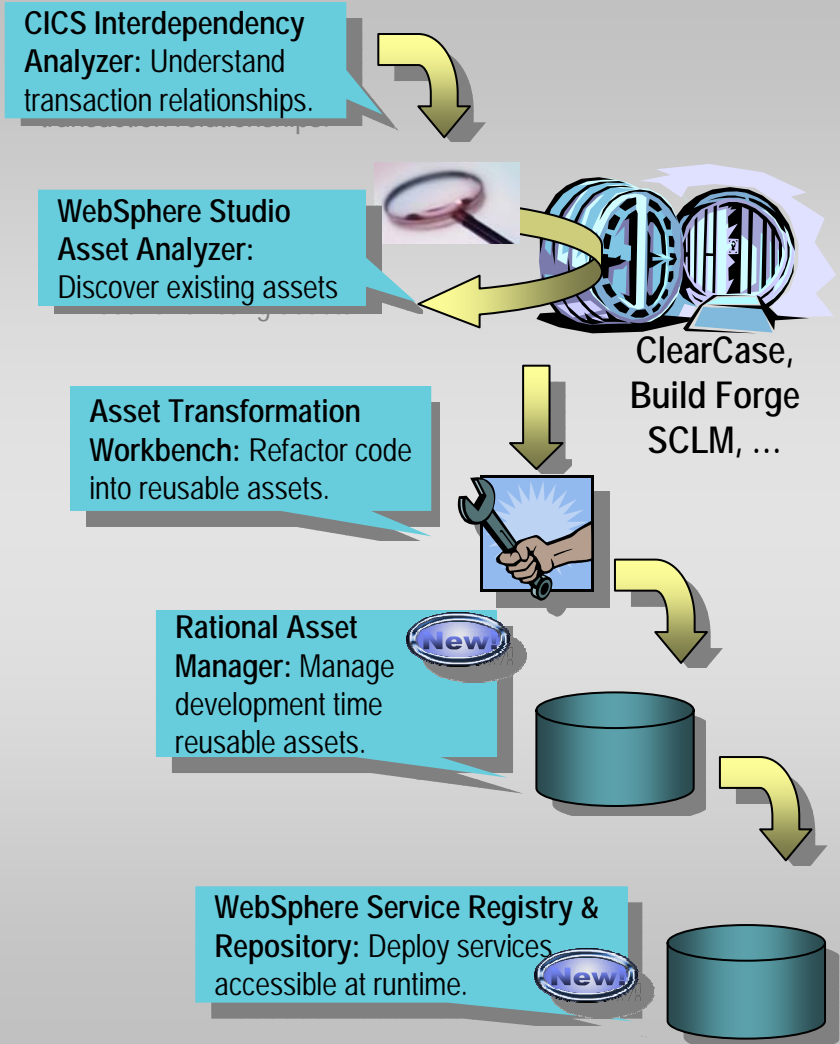
## Optimize development investments

### ► Scenario:

- We need to understand the impact of changes to existing code, and leverage it more fully. When building new assets / services, we need to publish them for reuse.

### ► What's here / coming:

- CICS Interdependency Analyzer – capture interdependency information and analyze transaction affinities
  - *New in v2.1:* Eclipse-based user interface to view runtime relationships
- WebSphere Studio Asset Analyzer – perform impact analysis across the enterprise
- Asset Transformation Workbench – perform pattern identification, extract business rules, assess suitability for reuse in SOA
  - *New in V2.1:* Reuse analyzer to identify potential services in COBOL code
- Rational Asset Manager – manage reusable assets during development
- WebSphere Service Registry & Repository – store, access, and manage info. about services



# Model Applications and Data

*Use model-driven development to create applications & services*

## ► Scenario:

- We want to architect services, applications and data to improve quality, flexibility

## ► What's here:

- Rational Software Architect – develop applications and web services using UML; use UML Profile for Software Services
- Rational Data Architect - help data architects design relational and federated databases, understand data assets and their relationships and streamline database projects  
*New for v7.0:* 1) Port to Eclipse 3.2 enables shell sharing between RDA and RAD, which results in a smaller footprint, less memory, simplified user experience, 2) expanded support for logical models

## Rational Data Architect

The screenshot displays the Rational Data Architect (RDA) interface within the IBM Rational Software Development Platform. The main window shows a hierarchical database model diagram with nodes for 'STDBANK', 'GENERATE', 'STANDARD', 'FIRSTNAME', 'SURNAME', 'MORTGAGE', 'ACCOUNT', and 'CORP\_OUT'. A 'Database Explorer' pane on the left shows the project structure, including 'Insurance logical model.idm' and 'Database'. The 'Properties' pane at the bottom right shows the definition for the 'AR\_ID' column:

Name	Primary Key	Domain	Data Type	Length	Scale	Not Null	Generated	Default Value/Generate E...
AR_ID	<input checked="" type="checkbox"/>		INTEGER			<input checked="" type="checkbox"/>		
PPPL_DT	<input checked="" type="checkbox"/>		DATE			<input type="checkbox"/>		
AR_TP	<input checked="" type="checkbox"/>		SMALLINT			<input type="checkbox"/>		
PPPL_ID	<input checked="" type="checkbox"/>		TIME			<input type="checkbox"/>		
FNC_SVC_PDA...	<input type="checkbox"/>		SMALLINT			<input type="checkbox"/>		
FNC_SVC_RSTC...	<input type="checkbox"/>		SMALLINT			<input type="checkbox"/>		
UNQ_ID_SRC_STM	<input type="checkbox"/>		CHAR	20		<input type="checkbox"/>		
RSTC_DPCLT_LV...	<input type="checkbox"/>		SMALLINT			<input type="checkbox"/>		
RSTC_IS_ATTRN...	<input type="checkbox"/>		SMALL INT			<input type="checkbox"/>		
FSWIC_CUST...	<input type="checkbox"/>		SMALLINT			<input type="checkbox"/>		
FSWIC_UTILZ_S...	<input type="checkbox"/>		SMALLINT			<input type="checkbox"/>		
FSWIC_RESPM...	<input type="checkbox"/>		SMALLINT			<input type="checkbox"/>		
PTRN_ESP_RNG_ID	<input type="checkbox"/>		SMALLINT			<input type="checkbox"/>		
CR_RNSI_S1_BI	<input type="checkbox"/>		SMALL INT			<input type="checkbox"/>		
FSWIC_UTILZ_S...	<input type="checkbox"/>		SMALLINT			<input type="checkbox"/>		
AR_PPPL_PMT...	<input type="checkbox"/>		SMALLINT			<input type="checkbox"/>		
PMT_PAID_NBR	<input type="checkbox"/>		SMALLINT			<input type="checkbox"/>		
PMT_PAID_NBR	<input type="checkbox"/>		SMALLINT			<input type="checkbox"/>		
AR_INQ_JRTH...	<input type="checkbox"/>		SMALLINT			<input type="checkbox"/>		

# Transform UML Models to Code

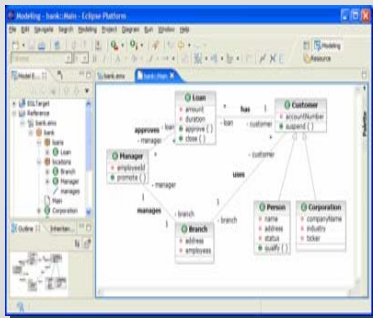
*Improve productivity and quality; transform from models to code*

## ► Scenario:

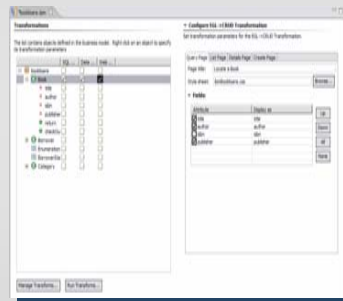
- We need to increase productivity and improve quality by transforming the models that our architects build directly into code for deployment to System z and/or other platforms

## ► What's coming:

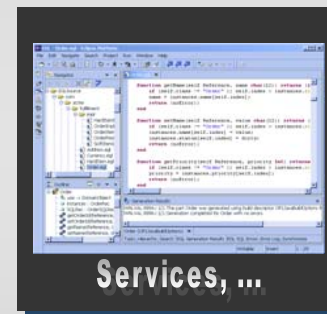
- Rational Business Developer Extension (RBDe), WebSphere Developer for zSeries – perform UML Transformations to EGL, COBOL, Web Services, XSD, C++



1. Model



2. Define Transformation Parameters



3. Transform to code



4. Deploy to platform (z, i, distributed)

- Traceability from requirements to code
- Create your own transformations
- Easily build / deploy Services on host



# Develop Applications

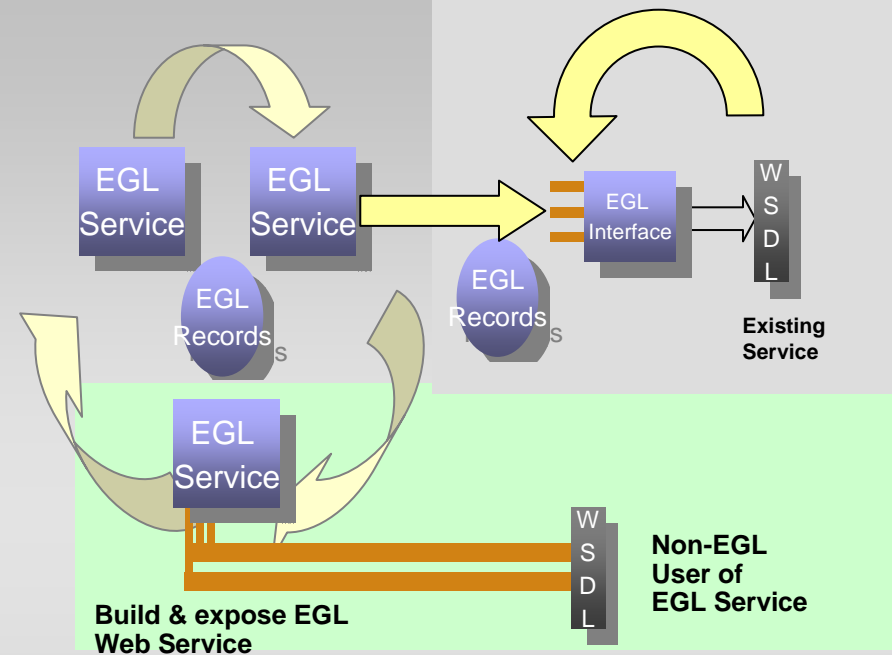
*Simplify creation of service-oriented applications for System z*

## ► Scenario:

- We need to create and maintain multi-platform applications where key portions run on System z platforms. We also need to extend our existing applications to leverage modern architectures like J2EE and SOA.

## ► What's here:

- WebSphere Developer for zSeries – accelerate the development of your Web (JSF/EGL), COBOL and PL/I applications, Web services; visually choreograph COBOL flows
- Rational Application Developer (RAD) - design, develop, analyze, test, profile and deploy Web, SOA, Java, J2EE and portal applications
- WebSphere Portlet Factory – rapidly create, customize, maintain, and deploy portlets



## ► What's coming:

- Rational Business Developer Extension (RBDe) – leverage Enterprise Generation Language (EGL) to implement SOA and achieve high productivity for CICS, IMS, z/OS Batch, WAS, System i, distributed



# Orchestrate Business Process Flows

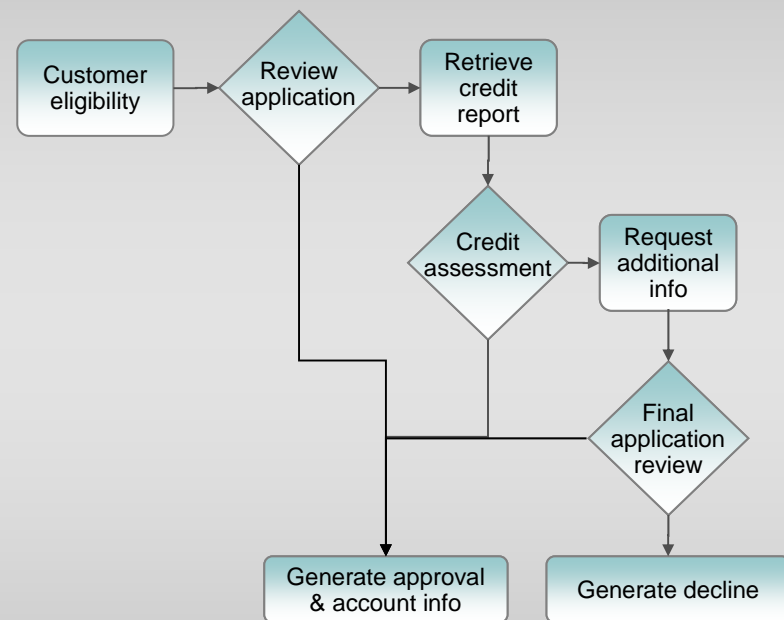
*Quickly assemble business solutions from reusable components*

## ▶ Scenario:

- We need to assemble business process solutions in a high-level, visual manner

## ▶ What's here:

- Websphere Integration Developer  
– visually describe your processes and rapidly assemble business solutions by wiring reusable service components



# Test & Debug Applications

*Save time and improve quality by debugging, automating, load testing*

## ▶ Scenario:

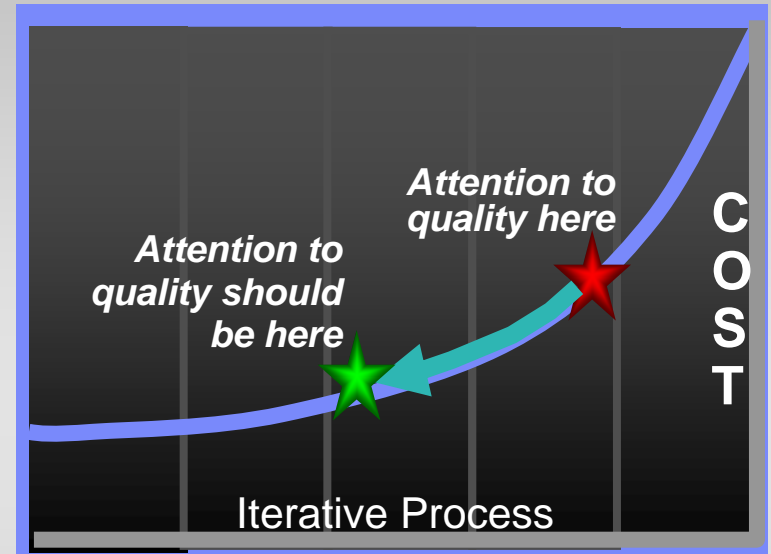
- We need a consistent testing methodology across all our applications regardless of platform. We also need help debugging applications in test.

## ▶ What's here:

- Rational Functional Tester – automate functional & regression testing
- Rational Performance Tester – validate application scalability before deployment
- Debug Tool Utilities and Advanced Functions – in conjunction with WDz, debug all components of a composite application

## ▶ What's coming:

- Rational Performance Tester – load test web services



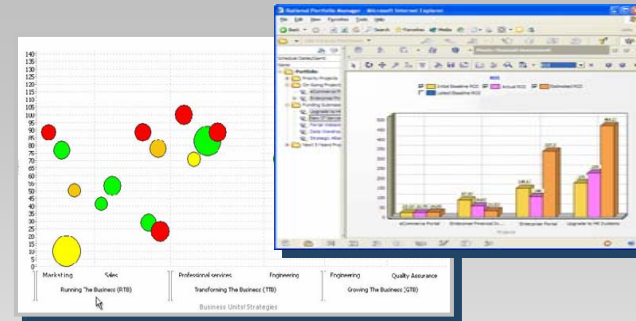


# Make Better Decisions

## Align business needs with development projects

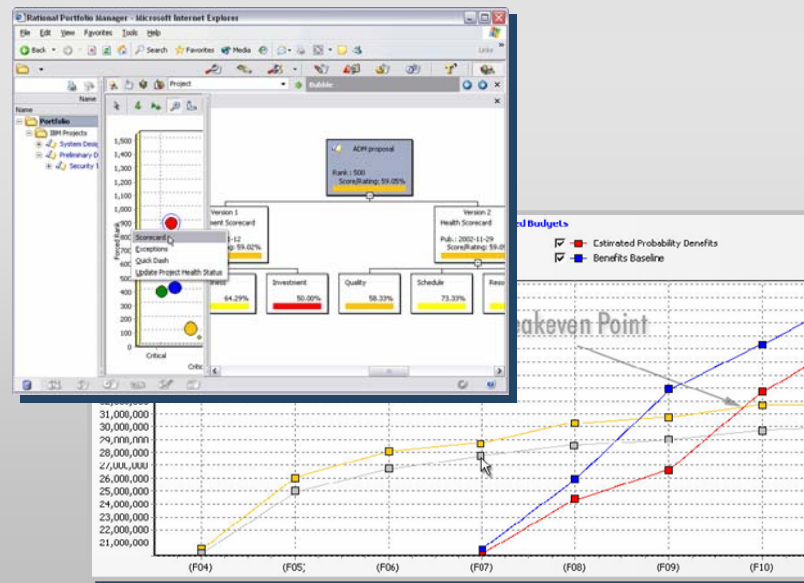
### ► Scenario:

- We need to better align the requirements of my business with the development projects we undertake, prioritizing and selecting the best projects for investment



### ► What's here:

- Rational Portfolio Manager – integrate with other SDP products, provide a unified dashboard
- Rational Method Composer – leverage, customize our best practices (RUP) that leverage IBM's expertise in portfolio management, collaborative distributed development, and service oriented architectures



### ► What's coming:

- RPM – use DB2 for z/OS

# Manage Source Code, Config Mgmt across the Enterprise

*Integrate configuration management, problem tracking for all platforms*

► **Scenario:**

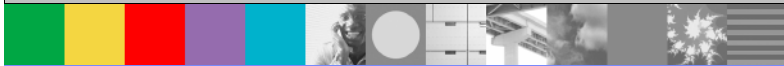
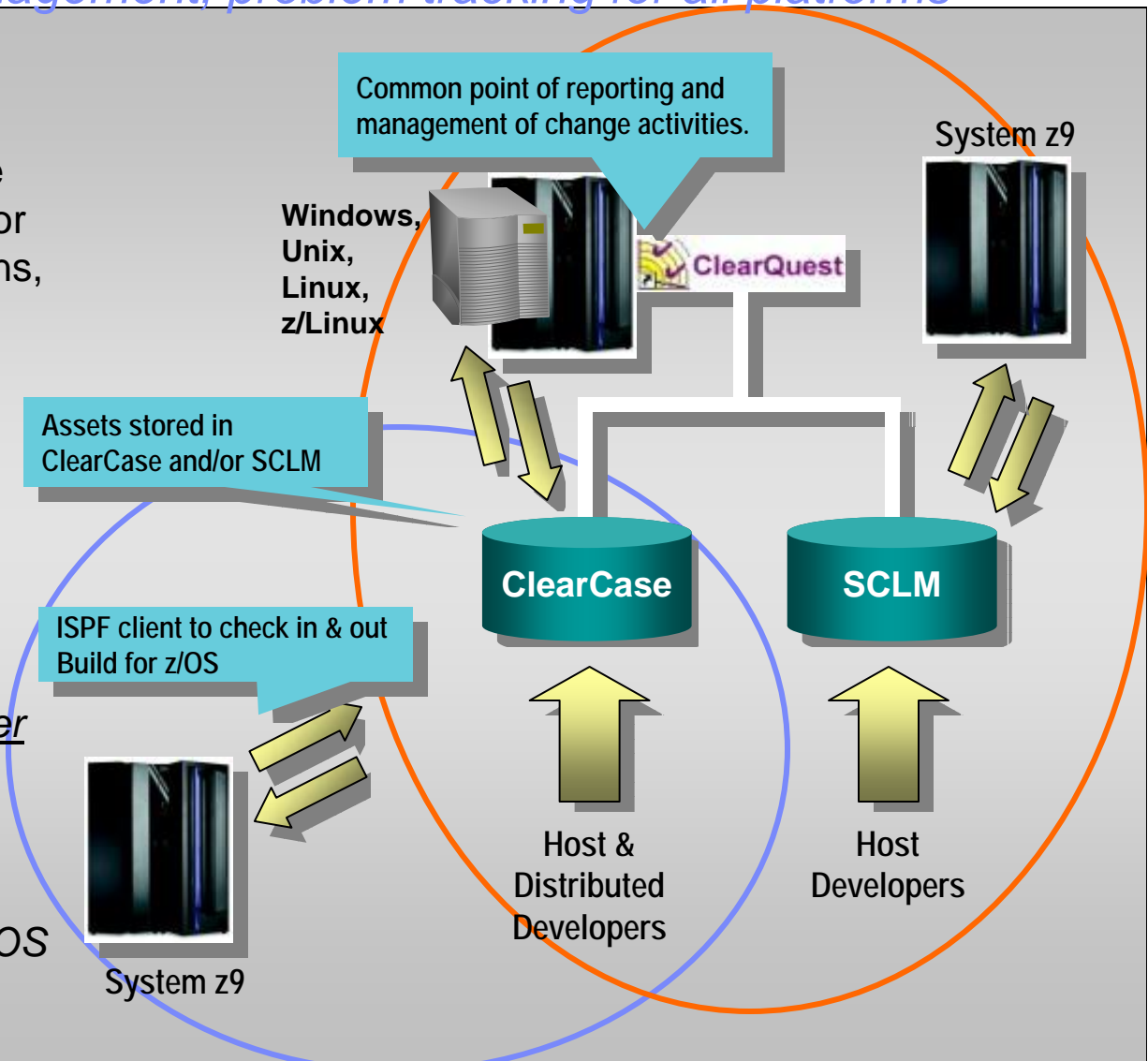
- We need a consistent source code management solution for all of our business applications, regardless of platform

► **What's here:**

- ClearCase - build for z/OS, ISPF client, run on z/Linux
- ClearQuest & SCLM - integration
- Source Code Library Manager (SCLM) - Eclipse client

► **What's coming:**

- ClearQuest – use DB2 for z/OS
- ClearCase – run on z/OS



# Manage and Automate the Build Process

*Get high performance, reliable builds throughout the development cycle*

## ► Scenario:

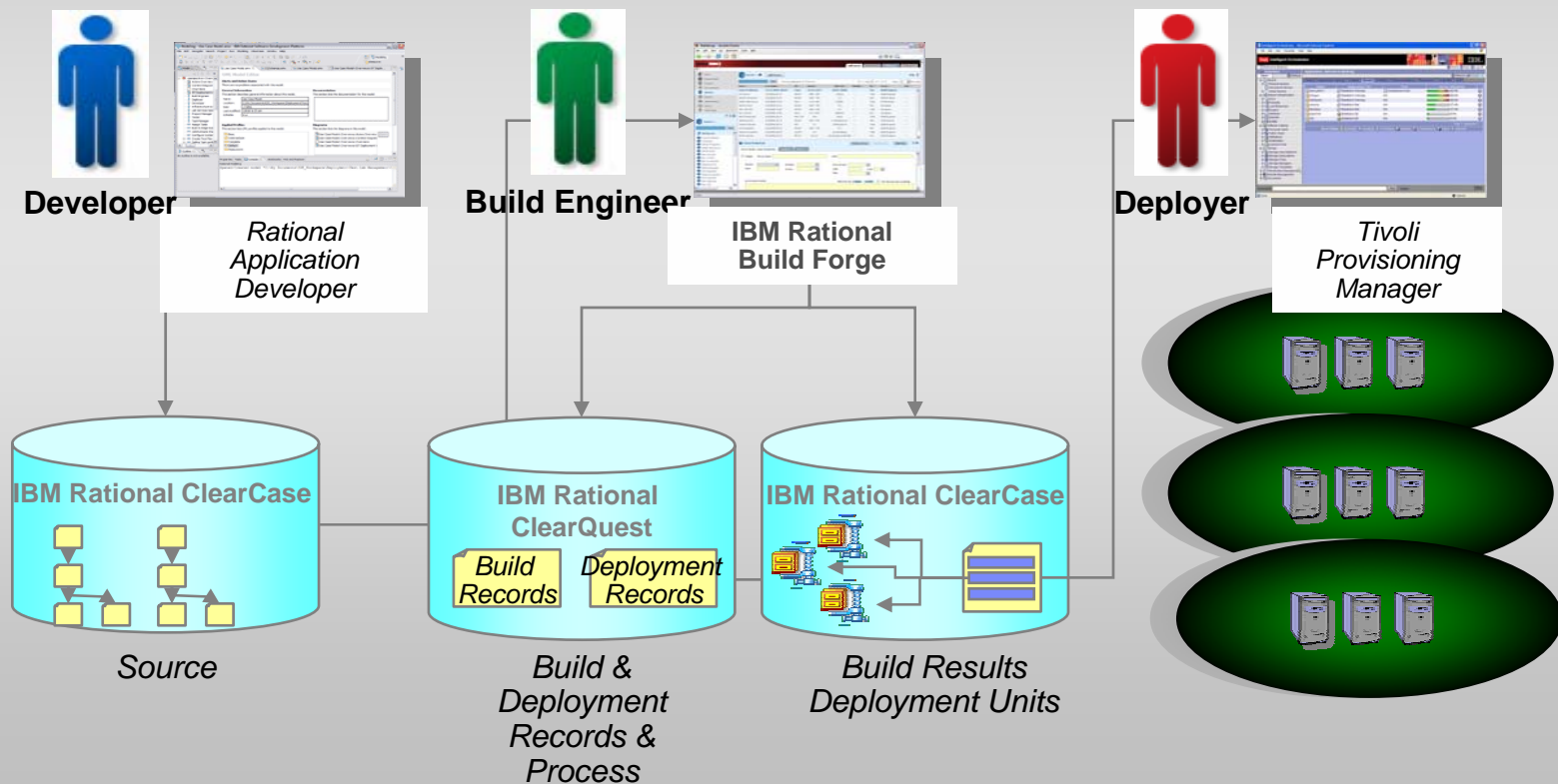
- We need to perform cross-platform builds throughout the development lifecycle.

## ► What's here:

- Rational Build Forge - perform coordinated builds across multiple platforms to streamline software delivery throughout the development lifecycle

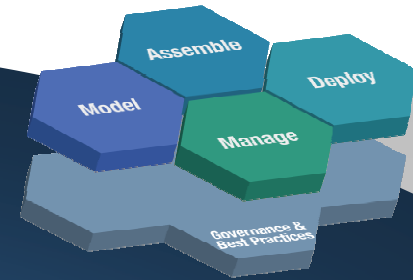
## ► What's coming:

- Rational Build Forge – z/OS build agent



# Share a Process across your Enterprise

*Leverage platform uniqueness across a common process*

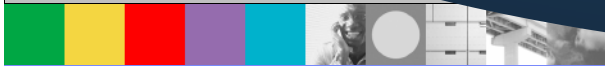
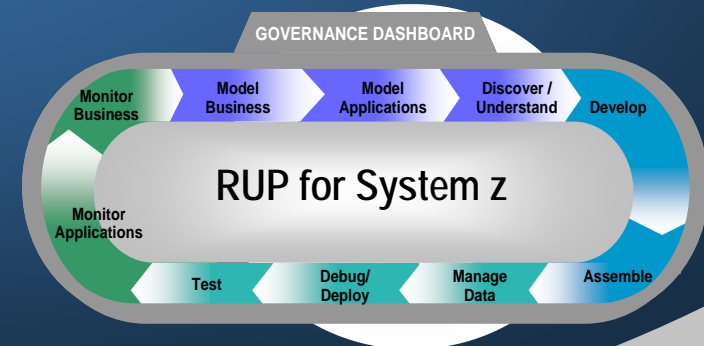
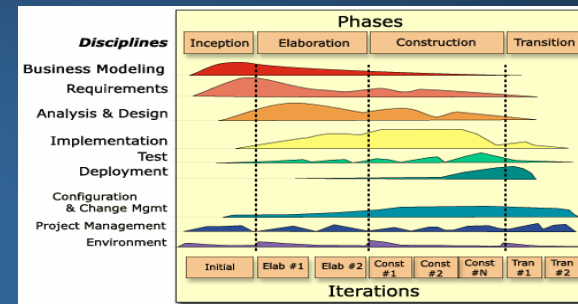


► **Scenario:**

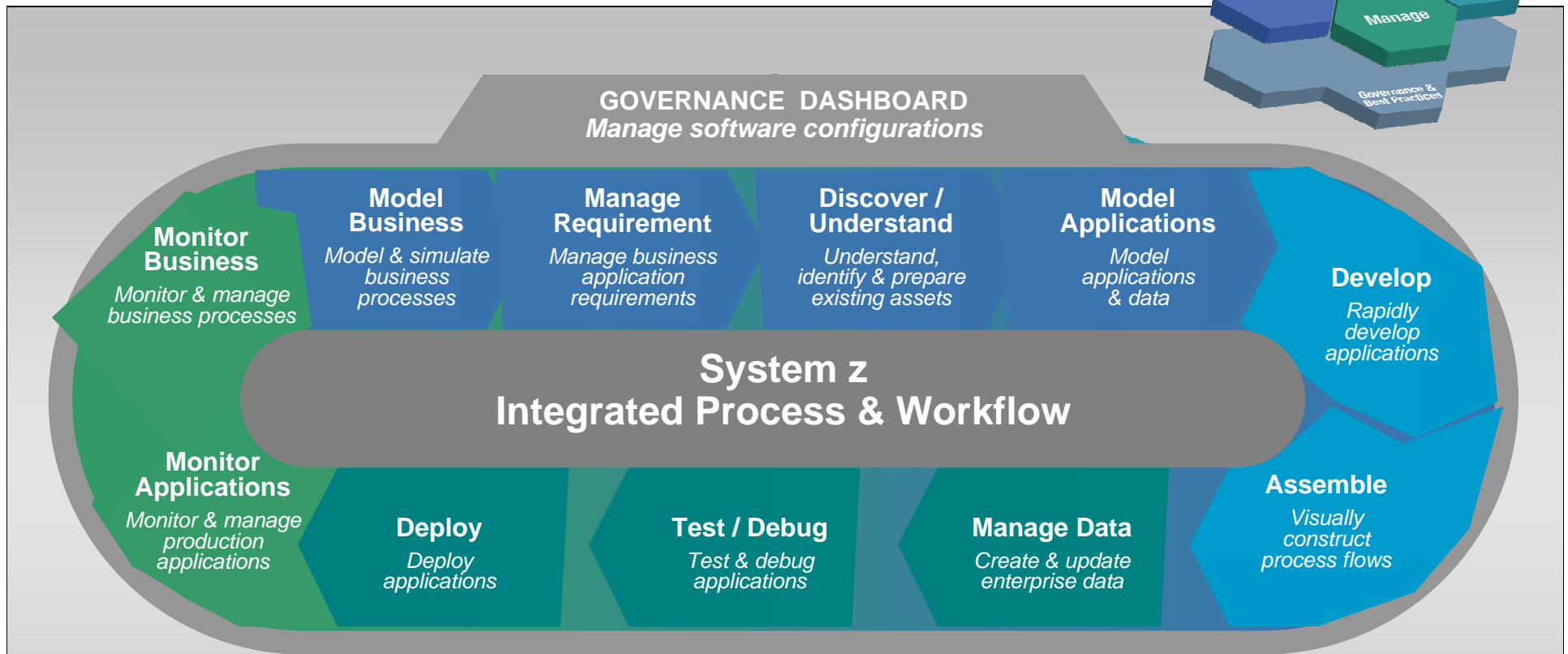
- We need a common development process across my enterprise that considers platform uniqueness

► **What's coming:**

- A specialization of RUP for SOA targeting development for System z
- Best practices for discovery, reuse and service creation from existing COBOL and PL/I code
- Integrates with
  - Rational Portfolio Manager for project execution and management
  - Eclipse-based development environments for non-intrusive practitioner guidance



# Software Development Lifecycle



▶ The IBM SDP provides

- Coverage of the complete software development lifecycle
- Products that support System z, System i, and distributed platforms via platform-specific and platform-agnostic support

## Summary

- ▶ The marketplace is undergoing rapid change; IT must adapt by leveraging:
  - Open Computing / Communities, Modularity and Good Governance
- ▶ Good governance
  - Creates business advantage
  - Empowers and enables practitioners
    - Process and governance should become part of the organization's culture and day-to-day work
    - A key enabler is process automation and information integrated into productivity tools
- ▶ The IBM SDP offers leading edge, high productivity tools for System z just as for distributed platforms.

Thank  
YOU



# Backup





# Deploy applications into production

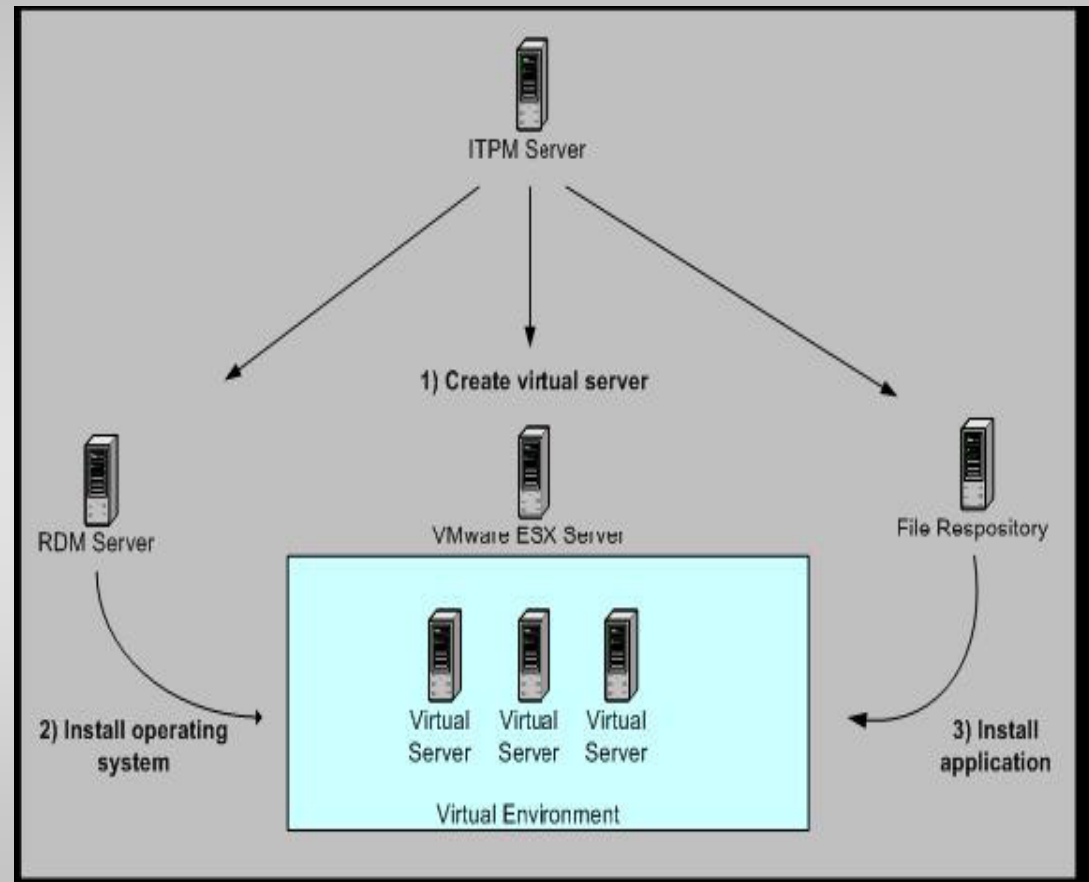
## *Simplify managing images across an enterprise*

### ▶ Scenario:

- We need to simplify and reduce the number of cloned images managed across our enterprise

### ▶ What's here:

- Tivoli Provisioning Manager – provide an easy-to-use console for remote deployment and management of operating systems



# Create & Update Enterprise Data

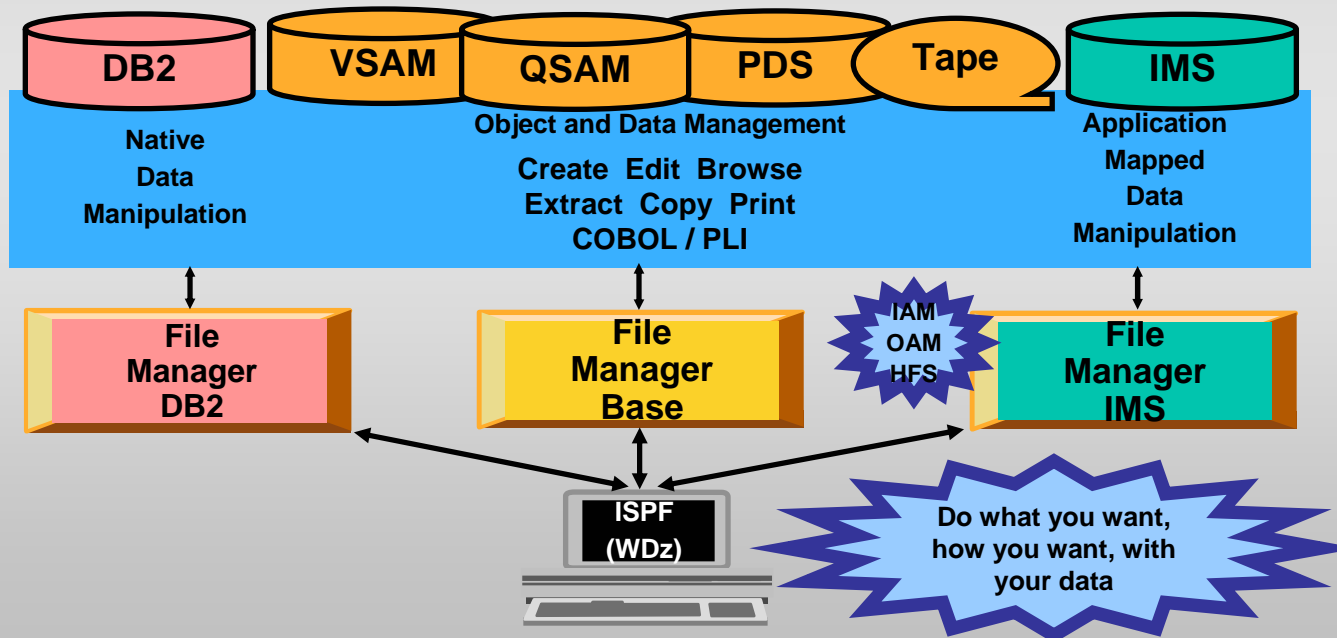
*Manage your enterprise data*

▶ **Scenario:**

- We need to manage a variety of enterprise data.

▶ **What's here:**

- File Manager for z/OS – manage data across multiple file formats and storage media
  - *New for V7:* generation of XML from files
- File Export for z/OS - export and import related sets of DB2, IMS, VSAM and sequential data



# Monitor and Manage Production Applications

*Keep your applications running*

## ► Scenario:

- We need a consistent mechanism for monitoring and managing all of our applications regardless of platform

## ► What's here:

- IBM Tivoli Composite Application Manager (ITCAM) - pinpoint the source of application bottlenecks
  - *New for V6: enhanced for SOA.*
- Application Performance Analyzer (APA) for z/OS – analyze, monitor, and report on performance
  - *New for V7: support for C/C++ application performance analysis and for DB2 Stored Procedures written in Java*
- Fault Analyzer for z/OS – pinpoint the cause of failed applications
  - *New for V7: continued fault analysis of latest versions of Websphere for z/OS and Java applications*

The screenshot displays the 'SERVER ACTIVITY DISPLAY' page in the ITCAM interface. At the top, there are navigation tabs: ADMINISTRATION, AVAILABILITY, PROBLEM DETERMINATION, PERFORMANCE ANALYSIS, LOGOUT, and HELP. Below the navigation is a 'SERVER SELECTION' section with a 'Group' dropdown set to 'Loan Application - CICS' and a 'Server' dropdown set to 'ADCDPL.M2L2.M2L2.CIC23QA2.bc3 (L3)'. There are two tabs: 'Active Transactions' and 'Recent Transactions'. The 'RECENT ACTIVITY (Last Minute)' table shows the following data:

SERVER INFO				RECENT ACTIVITY (Last Minute)	
Snapshot Date	Oct 12, 2004	Application Server Name	CIC23QA2	Process/Region CPU	2.70%
Snapshot Time	3:36:52 PM	Application Server IP Address	192.168.3.88	# of Transactions	0
Platform CPU % Utilization	4.00%	Total Task Count	1	Avg. Response Time (ms)	0
				# of Live Sessions	70

Below the table is the 'ACTIVE TRANSACTIONS' section, which includes a filter by 'Task Type' (Any) and 'Task Status' (Any), and a 'Refresh' button. The active transactions table shows:

Transaction	Transaction Start	Task ID	Resident Time (ms)	Accumulated CPU(ms)	Idle Time (ms)	Task Status	Current Program	Current Action
CSMI	Oct 12, 2004 3:22:54 PM	75	1050598	3.536	1050594	Waiting	DPLCIC2A	EXEC CICS Reque