



zEnterprise – An Ideal Basis For Smarter Computing

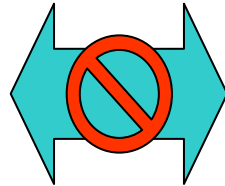
Developing Hybrid Applications For
zEnterprise

Smarter Computing Means Breaking Down Cultural Boundaries That Inhibit Optimum IT

Distributed teams



Mainframe teams



- Cultural barriers preclude fit for purpose optimizations
- Separate teams produce separate solutions
- Different skills inhibit optimum use of human resource

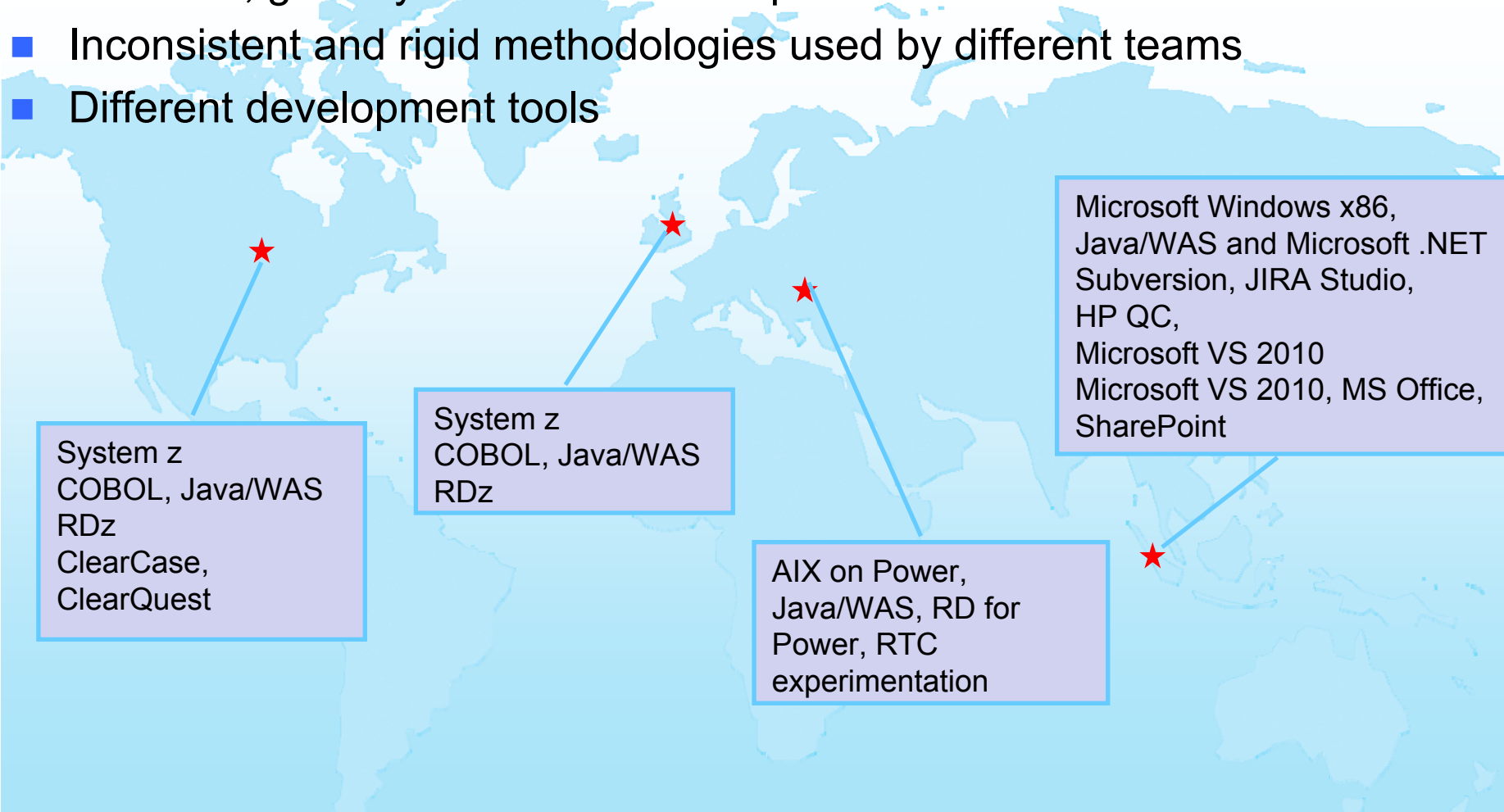
zEnterprise enables cultural integration



- Consolidate development and test around common tool set
- Optimize development process
- Reduce costs and overhead

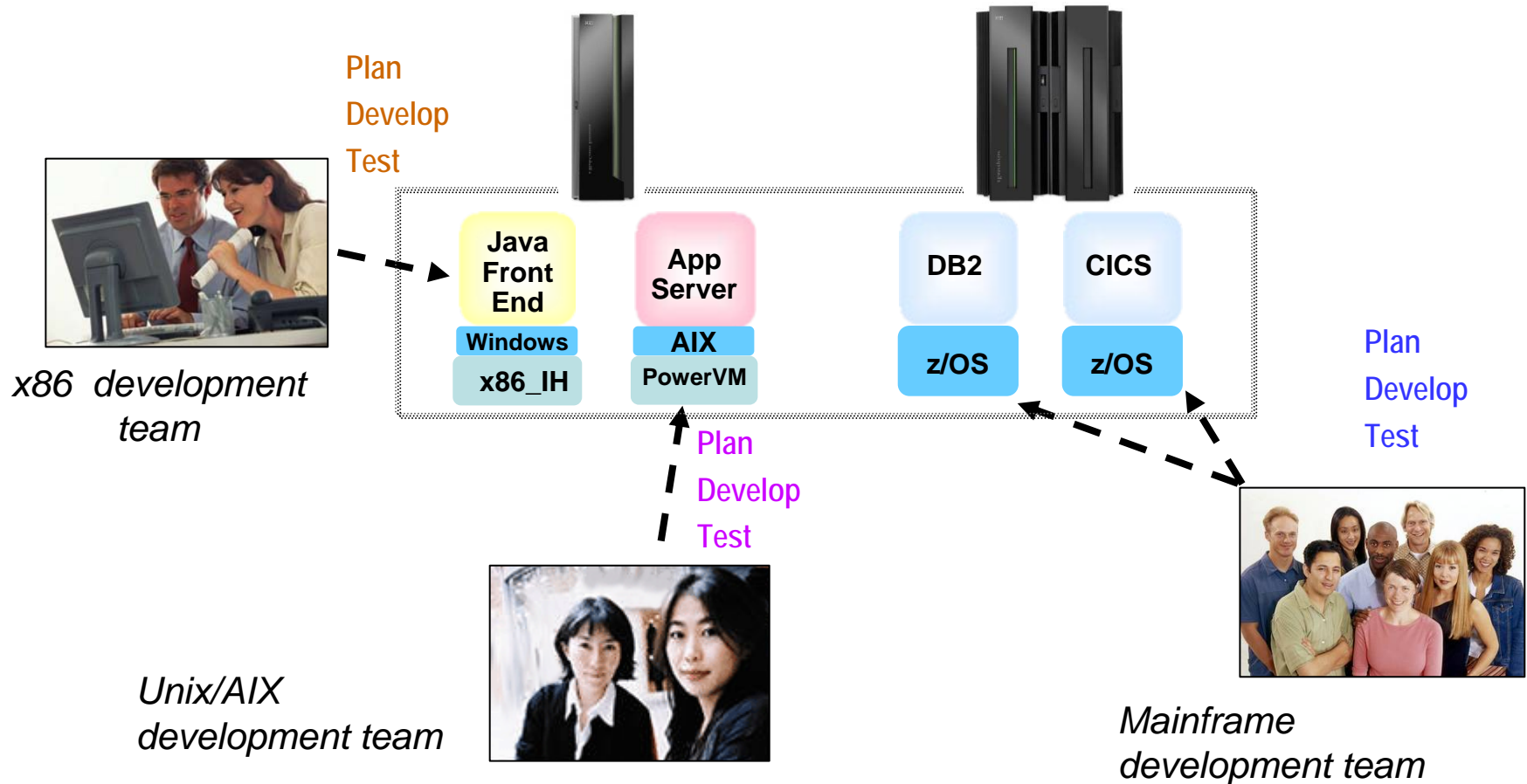
Development Infrastructure And Approach Can Contribute To Development Inefficiencies

- Multi-site, globally distributed development
- Inconsistent and rigid methodologies used by different teams
- Different development tools



How Will These Different Teams Productively Coordinate A zEnterprise Solution?

Typical 3-tier Web-facing application:



Barriers Cause A Breakdown In Communication

Analyst: “People don’t seem to be on the same page.”



Stakeholder: “I do not have any time or resources, so we will have to catch up later.”



Project Manager: “Do I have full coverage? What is the live status of the work?”



Quality Professional: “What requirements am I working on? Is anyone seeing my defects?”



Current issues...

- Our methodologies and processes aren’t working together
- Our worldwide teams’ tools don’t integrate
- We have no consolidated hub of information
- Local differences are hurting productivity



Developer: “Where did this defect come from? Who is testing my code?”

Without Collaboration, Development Has Problems

- Only 37% of stakeholders are satisfied with the speed of internal application development¹...
- Only 42% are satisfied with the quality¹...
- 50% of outsourced projects *expected* to underperform²...

WHY?

Geographic Barriers

- Poor communication
- Language, culture, time
- Process gaps resulting in rework
- High degree of friction

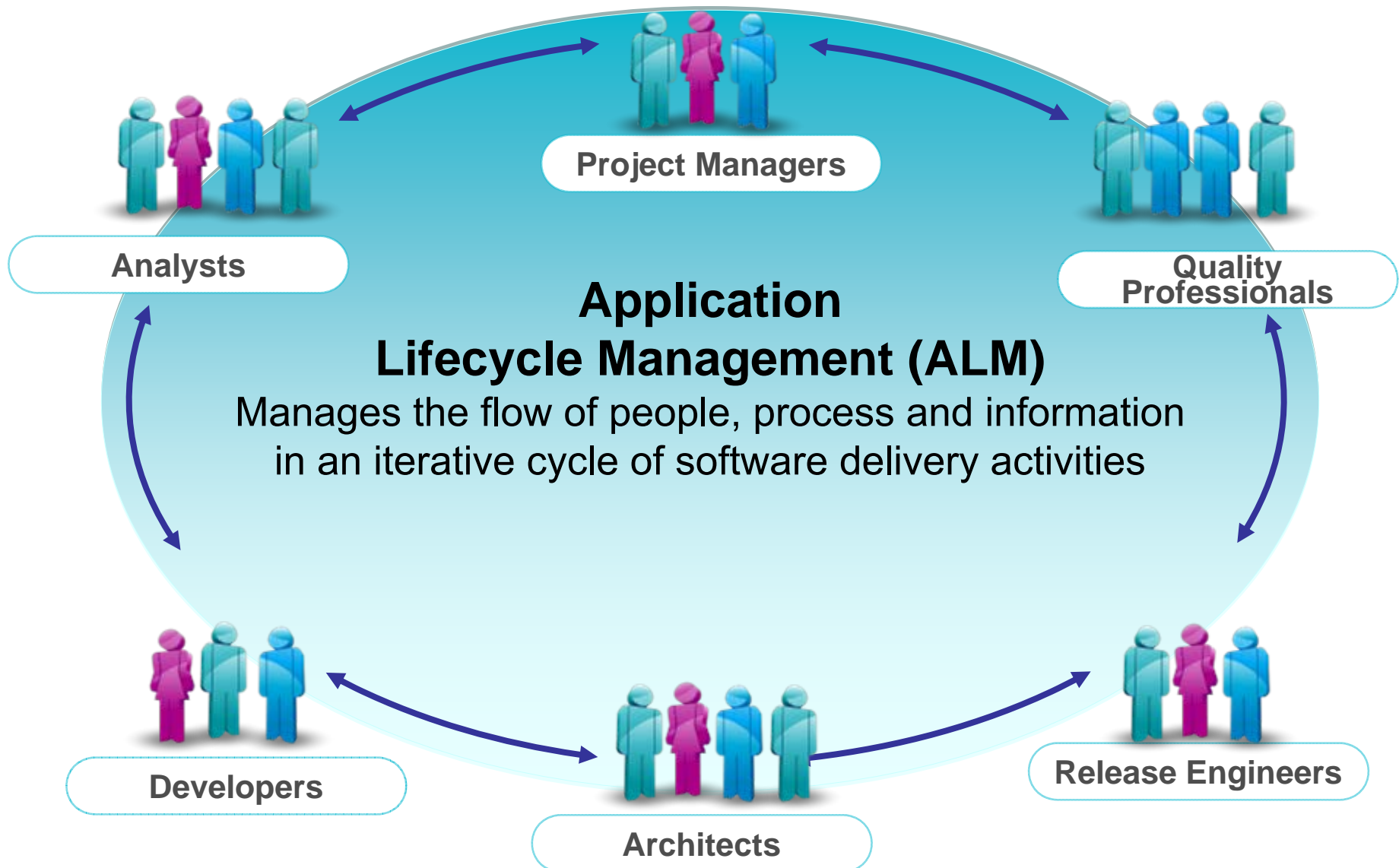
Organizational Barriers

- No meaningful collaboration
- Weak project governance
- Lack of domain expertise
- Poor LOB oversight
- Security when outsourcing

Infrastructure Barriers

- Incompatible tools / repositories
- Unreliable access artifacts
- Lengthy on-boarding
- Inflexible tooling integration

IBM Rational Adopts A Whole-team, Whole View Approach To Software Delivery



Rational Jazz-Based ALM Tools Are Used By All Team Members

Seamlessly integrated tools

**Rational
Requirements
Composer**

RRC

**Analysts
Project Managers
Architects
Development Managers
Testers and Test Managers
All Stakeholders**

**Rational
Team
Concert**

RTC

**Architects
Developers
Development Managers
Build&Release Engineers
Project Managers
Testers and Test Managers
Analysts**

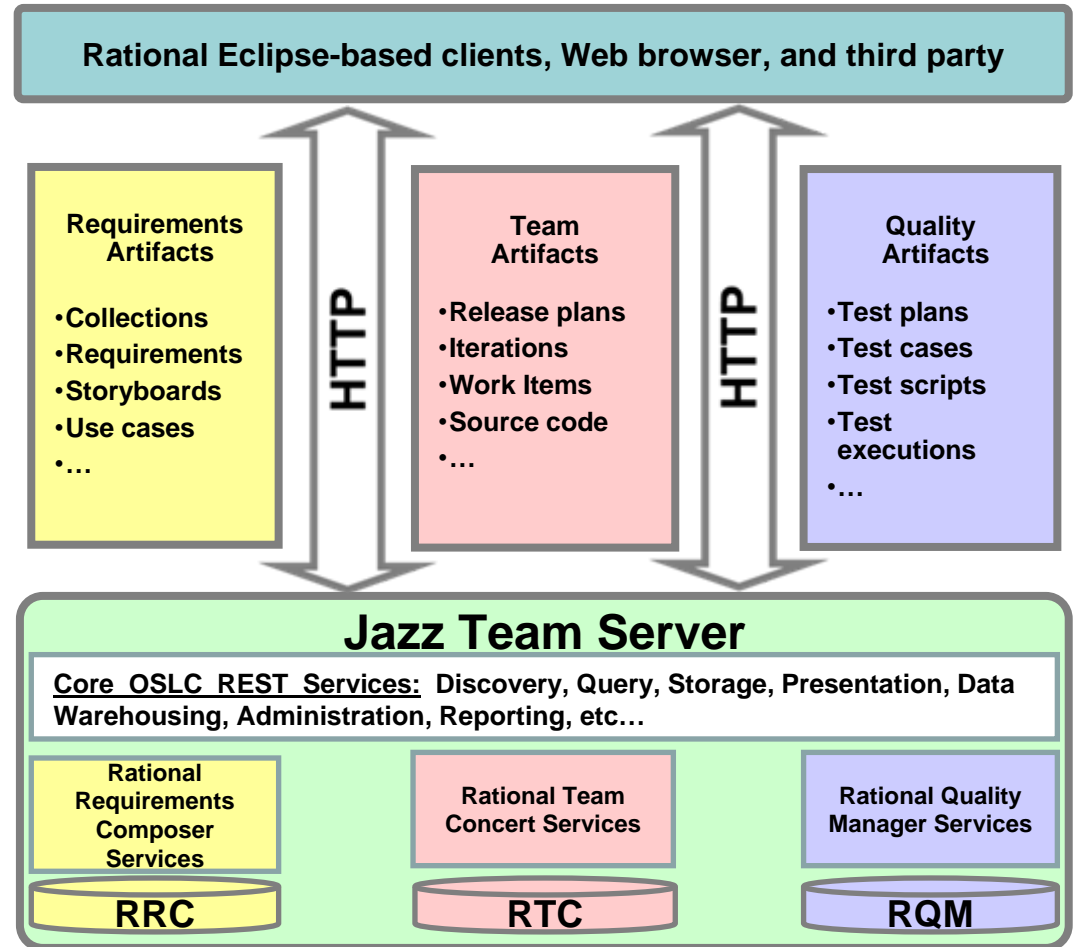
**Rational
Quality
Manager**

RQM

**Testers
Test Managers
Analysts
Project Managers
Development Managers**

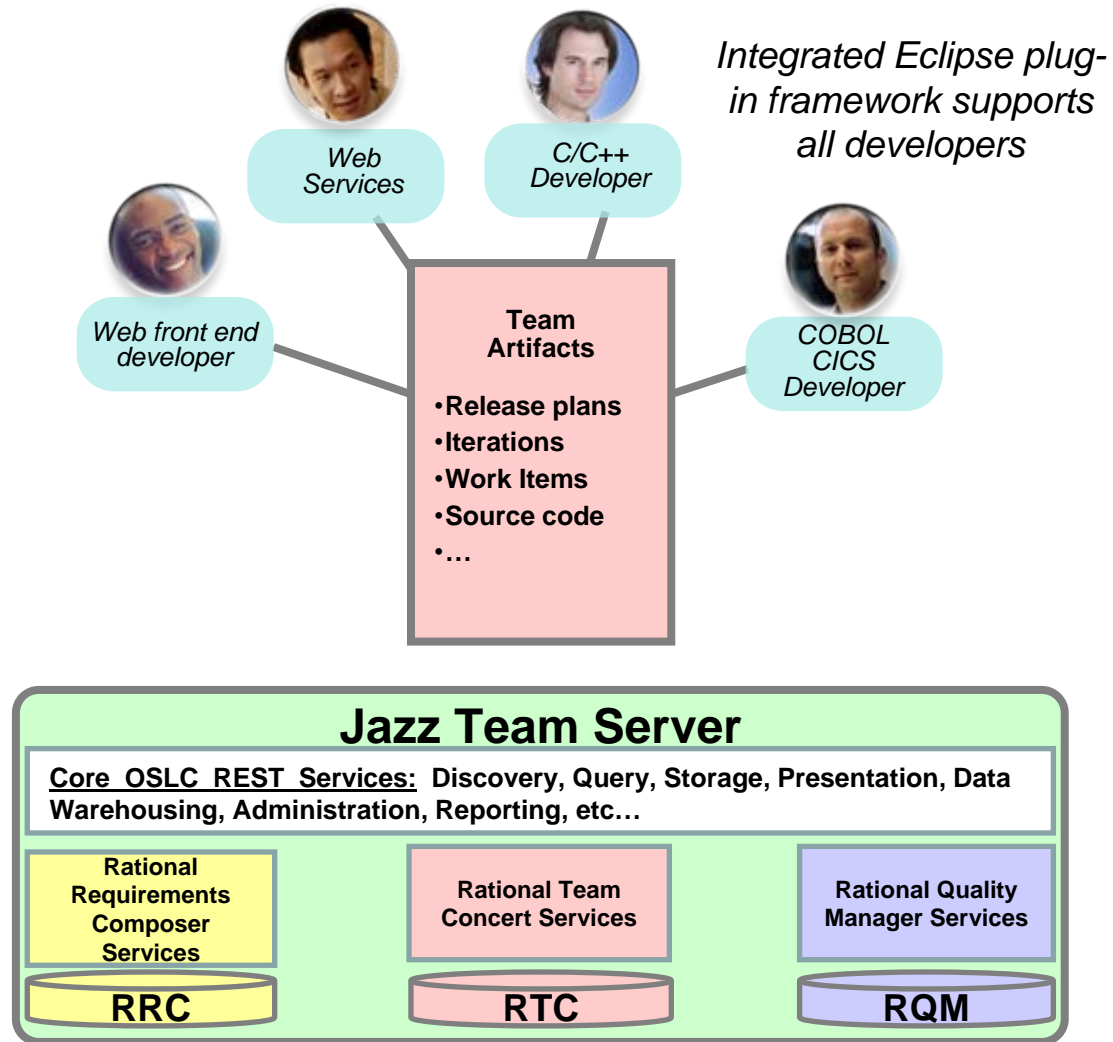
Jazz Foundation Breaks Down Development Barriers

- Jazz Team Server provides common services
 - ▶ E.g., shared data, collaboration tools, query, administration, etc.
 - ▶ Comprehensive Application Lifecycle Management
- The tools are built on Jazz Team Server:
 - ▶ Rational Requirements Composer
 - ▶ Rational Team Concert
 - ▶ Rational Quality Manager
- A single and consolidated hub of information
 - ▶ All artifacts are shared resources addressed by URLs
 - ▶ Accessible via REST API
 - ▶ Open Services for Lifecycle Collaboration (OSLC)



Integrated Development Environment Means Common Tools For All Platforms

- Develop cross-platform hybrid applications using *integrated* tools
- Rational Team Concert development tools contained within different “perspectives” in Eclipse environment
 - ▶ Java, Web and script developers use **Rational Application Developer (RAD)**
 - ▶ Traditional mainframe developers use **Rational Developer for System z (RDz)**
 - ▶ Unix / AIX developers use **Rational Developer for Power Systems (RDp)**
- RTC promotes collaboration across developers



Mainframe Programmers Can Continue To Develop Using Traditional Tools If Desired

- Traditional ISPF programmers can continue to use familiar green-screen interface...
 - ▶ **ISPF Client for Team Concert**
- ... but can integrate with Rational team services for software change management (SCM) functions
 - ▶ Use repository workspaces, change sets, link to work items, build requests, etc.
- Check out/check in code to native z/OS file system
- Facilitates phased implementation
- Reduces dependency on RDz deployment

```
Menu Help
RTC/z Primary Option Menu
Option ==> 2
0 Settings Terminal & user parameters ***** Logged in *****
1 Connection Work with Connection to source Userid . . : robin
2 Workspaces Work with repository Workspace Language. : ENGLISH
3 Edit Work with source data Server . . :
4 Build Work with Build options Project . . :
X Exit Terminate RTCz Workspace :
Release . :
```

ISPF SCM Client

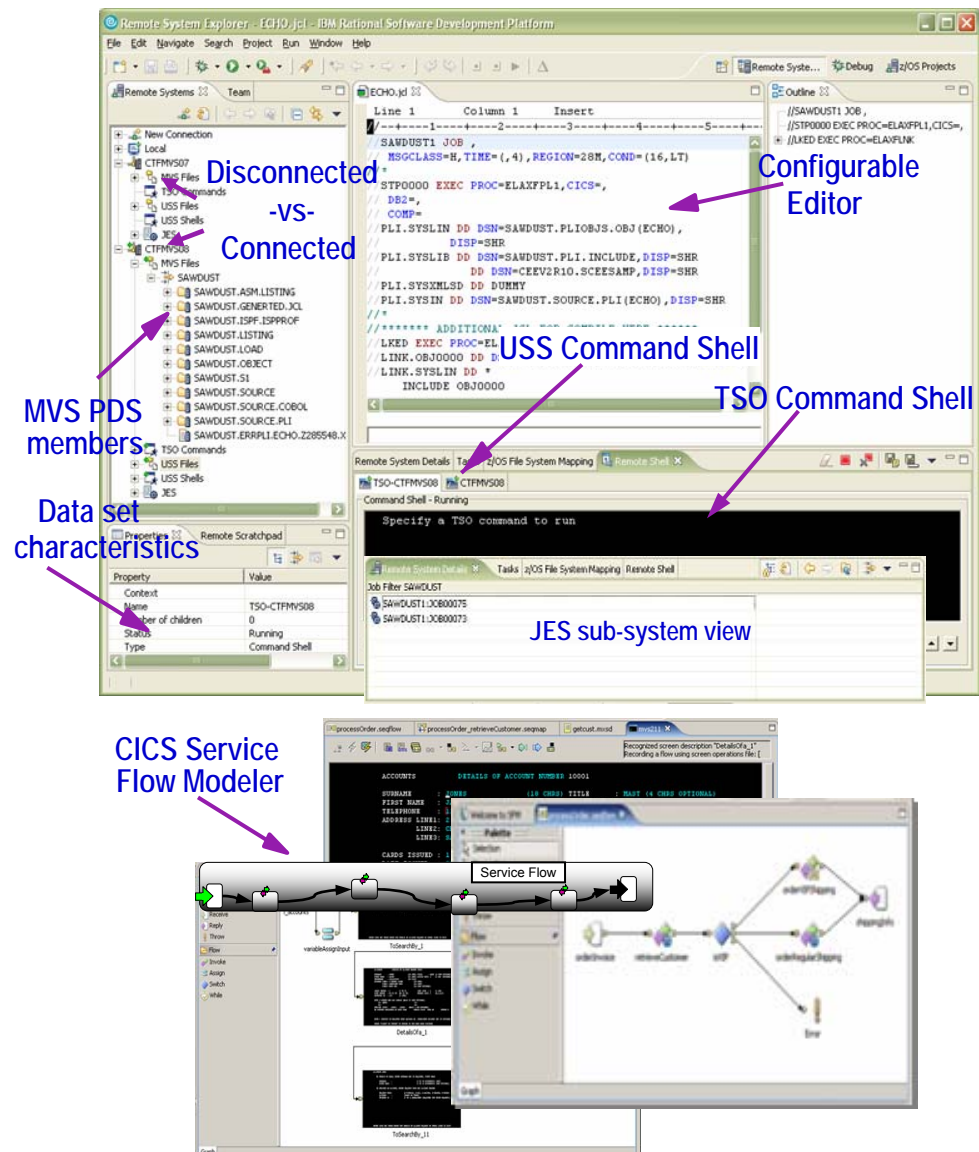
```
Menu Utilities Help
Repository Workspaces Row 1 to 3 of 3
Option ==> Scroll ==> CSR
Enter new repository workspace name to create or "/" against existing
repository workspace for options
Names Load location
> Mortgage App Dev USER55.SANDBOX
Test Workspace
Weekly Integration Workspace
***** Bottom of data *****
```

Summary Of Jazz Benefits

- Enables companies to break down geographical, infrastructure, and organizational barriers
- Collaborative Application Lifecycle Management provides transparent, end-to-end, governable solution to handle all distributed processes and methodologies
- Rational Team Concert provides a consolidated hub of all assets for requirements, change and release, and quality management teams
- Development tools are optimized for target platforms, including System z
- Competitors don't have the breadth of offerings to provide complete application lifecycle management like IBM

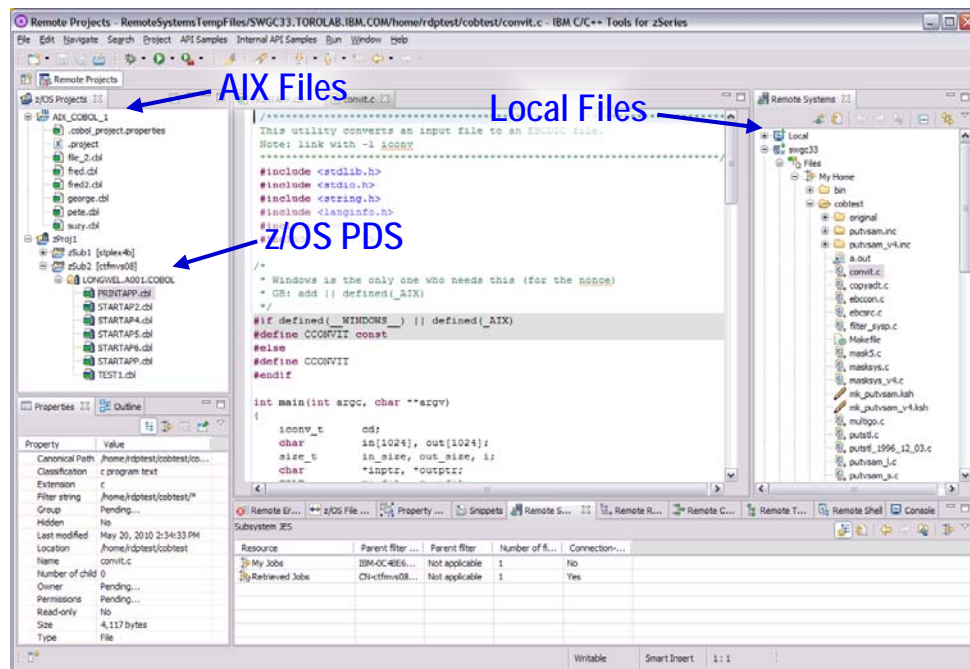
Rational Developer for System z Optimizations

- Mainframe developers move to a graphical integrated development environment
 - ▶ **Rational Developer for System z** provides full support for development and reuse of all mainframe assets
- Support for COBOL, PL/I, C, C++, HLASM, Java, EGL and Web services
- Supports existing and new runtimes
 - ▶ CICS, IMS, Batch, USS, DB2, WAS
- Interactive access to z/OS for debug, job generation, submission, monitoring, command execution, etc.



Rational Delivers Integrated Development For zEnterprise Solutions

- Specifically designed for solutions development on zEnterprise
 - ▶ **Rational Developer for zEnterprise**
- Combines the functionality of z, Power Systems, x86 and applications development
- Addresses unique capabilities and requirements of zEnterprise
- Includes end-to-end debugging across all environments
- Lowers the cost of traditional mainframe application development
 - ▶ Uses selective workload offloading
 - ▶ Reduces MIPS used for common dev activities



Work with artifacts on multiple platforms in one GUI

Studies Show Rational Developer For System z More Productive For Developing z/OS Applications

Comparison of Rational Developer for System z to ISPF:

Task	Test Results
Build a traditional CICS/COBOL/DB2 application	RDz was 1.2x faster
Enable CICS applications for Web Services	ISPF could not complete the task
Compile, test and debug	RDz was 1.2x – 1.7x faster

Conclusions:

- ✓ RDz was more productive for building robust real-world mainframe and Web based applications
- ✓ RDz was more productive at meeting applications requirements with minimum amount of tools

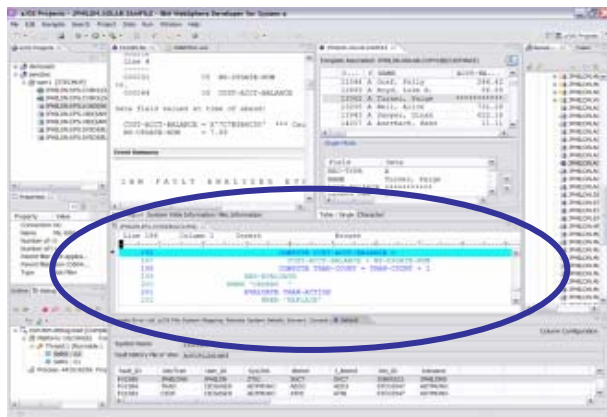
Source: The Branham Group, Inc.

<http://www.branhamgroup.com/CustomContentRetrieve.aspx?ID=2002290&A=SearchResult&SearchID=2196098&ObjectID=2002290&ObjectType=35>

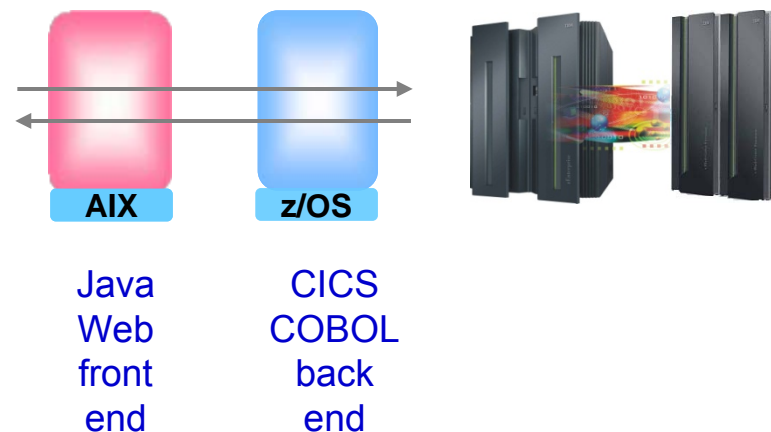
Hybrid Multi-tier Applications Are Easily Debugged

- Rational developer tools include integrated debuggers
 - ▶ Debug and step across languages
 - ▶ Debug and step across environments
- Collaborative debugging when RTC is integrated

Work with code in debugger on workstation

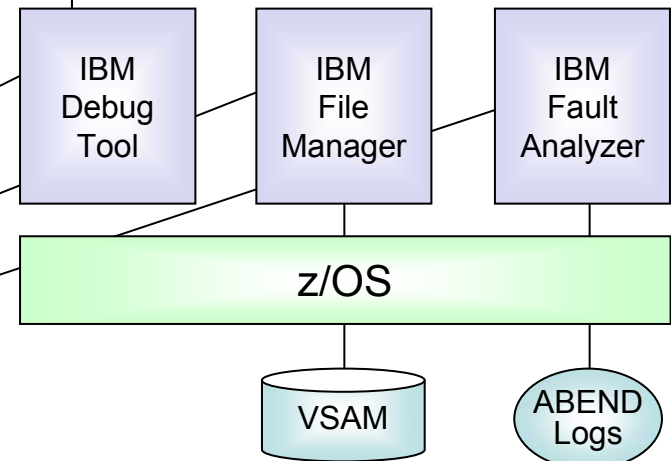
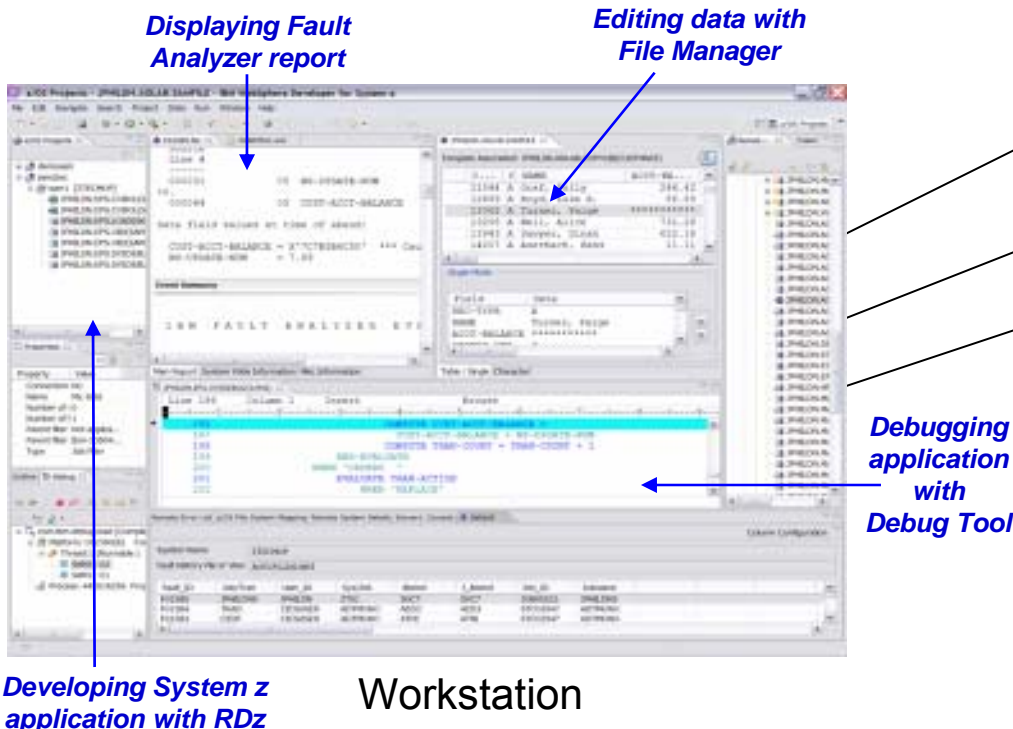
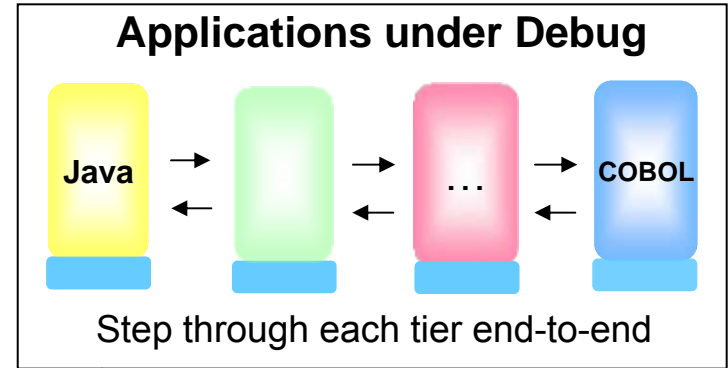


Debug applications running on all zEnterprise platforms



Debugging Includes Integration With Mainframe Problem Determination Tools

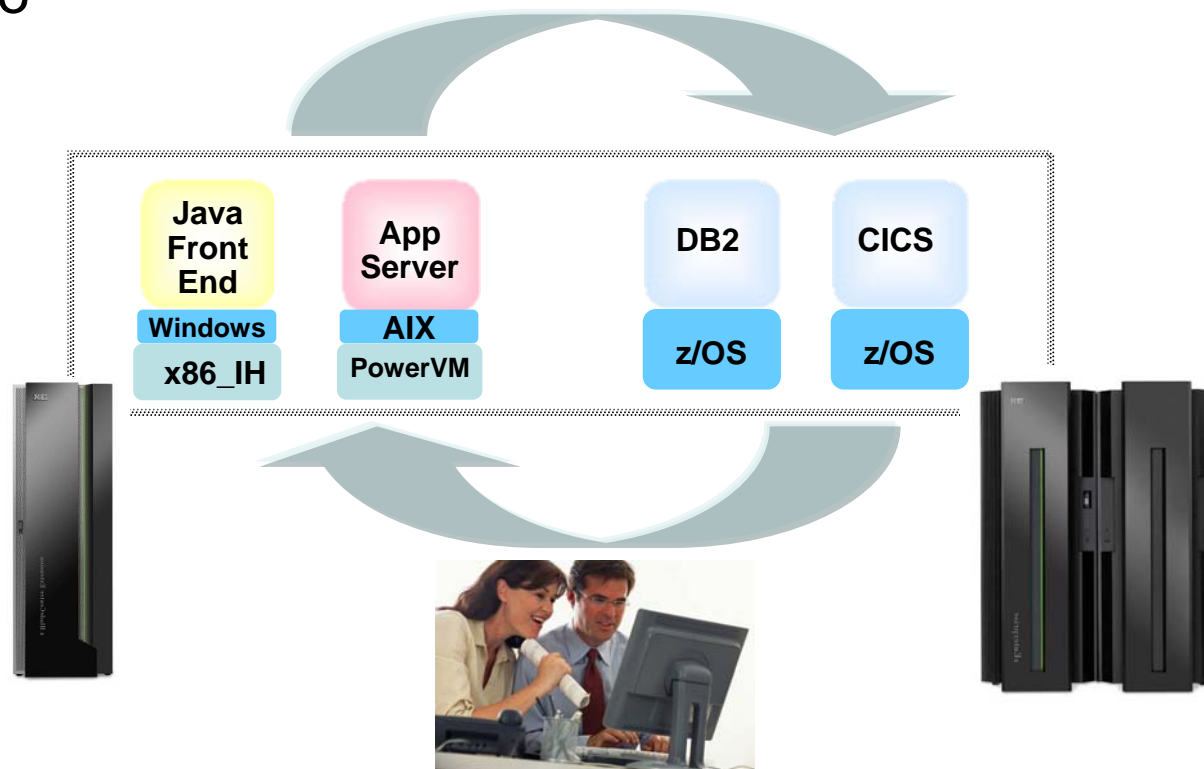
- Work with the PD Tools through the RDz client
- Easy access to all PD tools at the same time
- Debug and step through multi-tier applications
 - ▶ Across distributed *and* mainframe
 - ▶ With the same debugger



- End-to-end debug
- Edit VSAM data
- Analyze ABEND logs!

DEMO: End-To-End Debugging Of A Typical Multi-tier Application On zEnterprise

- Example of end-to-end debugging
 - ▶ Start in middleware tier (JSP)
 - ▶ Step through to COBOL tier
 - ▶ Step back to beginning tier



Testing The Full Extent Of A Multi-tier Application Is Critical For Code Quality

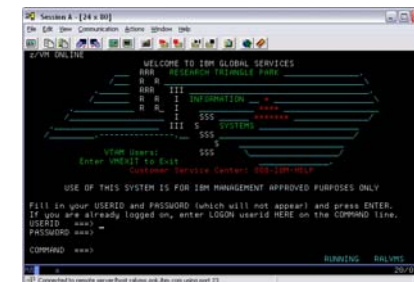
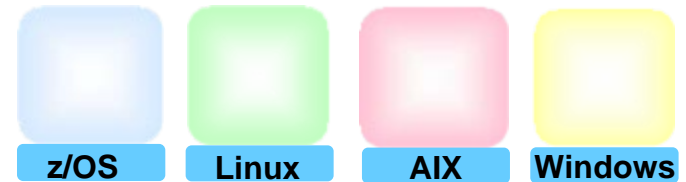
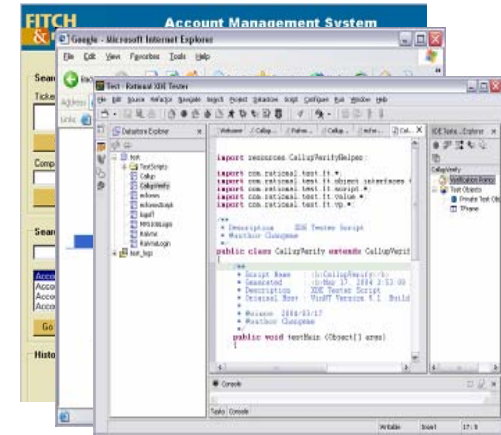
- Application quality is measured at many levels
 - ▶ Unit test, functional test, system test, performance test, etc.
- Quality needs to extend to all platforms (Mainframe, Power, System x, etc.)
- Test procedures need to seamlessly step across platforms for complete end-to-end debug
- Need to continue to use existing System z problem determination and debug capability...

... but it's important that the cost of testing be reduced if possible

Test All Aspects Of A Hybrid Application With Integrated Quality Management Tools

- Manage tests with Rational Quality Manager
- Record and playback scripts on Windows or Linux to functionally test a .NET, Web, Java and other application (z or non-z)
 - ▶ Rational Functional Tester
 - ▶ Rational Functional Tester Extension for Terminal-based Applications
- Performance test any Web application (z or non-z)
 - ▶ Develop scripts on Windows/Linux and execute scripts on z/OS
 - ▶ Rational Performance Tester for z/OS
 - ▶ IBM Workload Simulator for z/OS and OS/390 to test terminal-based applications

Web and GUI Applications



System z Terminal UI

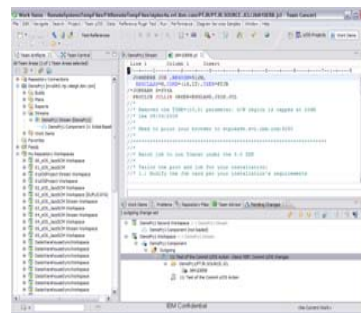
Unit Test Option For z/OS Applications Can Reduce Testing Costs

- z/OS runtime environment runs on x86 Linux workstation
 - ▶ Compile and unit test on the workstation
 - No zEnterprise hardware needed
 - ▶ Emulates System z general purpose processors, zIIPs, and zAAPs
- Reduces development MIPS for z/OS applications
 - ▶ Lower cost, better productivity and enable new skills quickly
- Includes latest compilers, middleware, server load modules for RDz & RTC
- Also available for educational institutions

**Rational Developer
For System z Unit
Test**



**Edit/Compile
Unit Test**



RDz

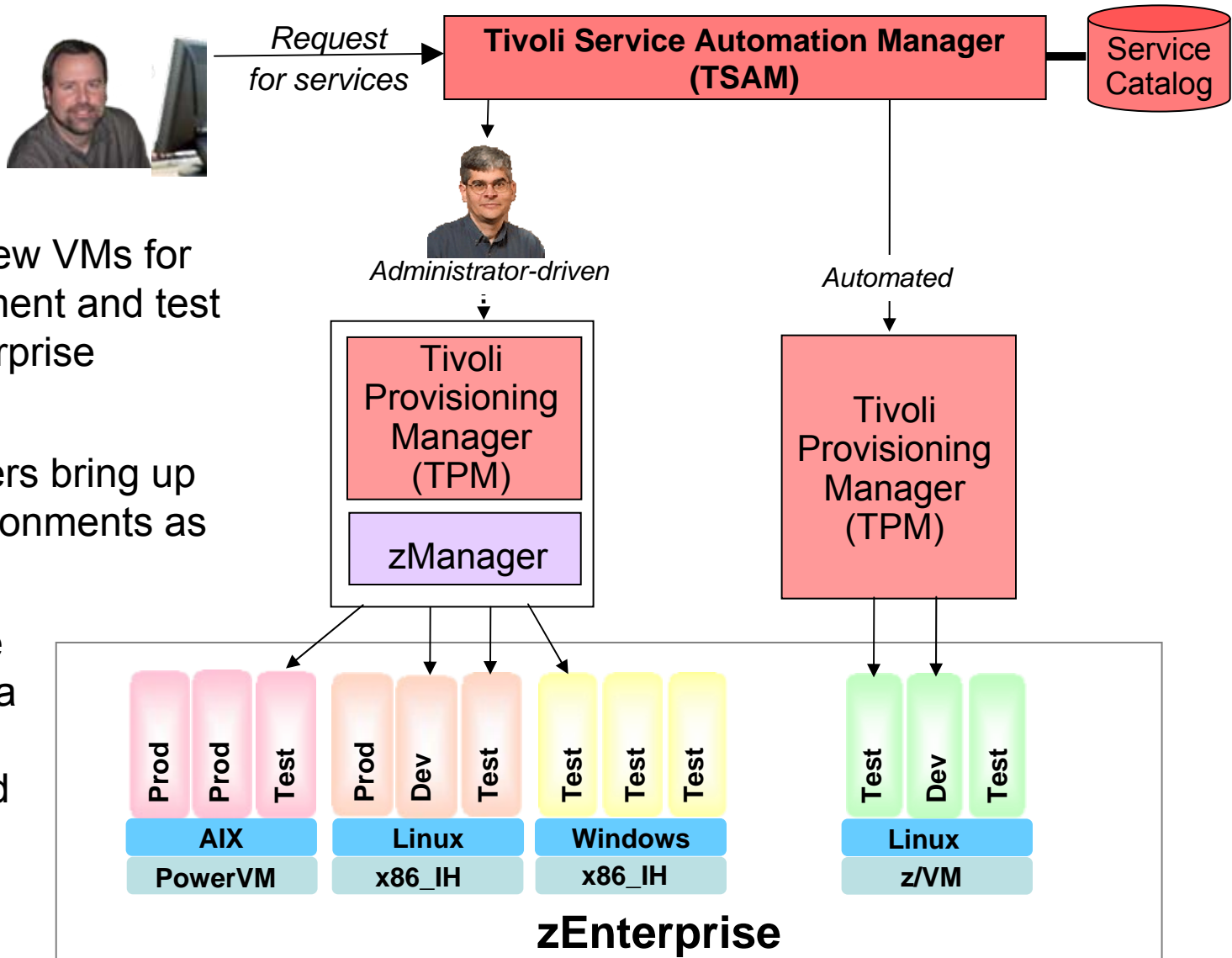
Run



z/OS

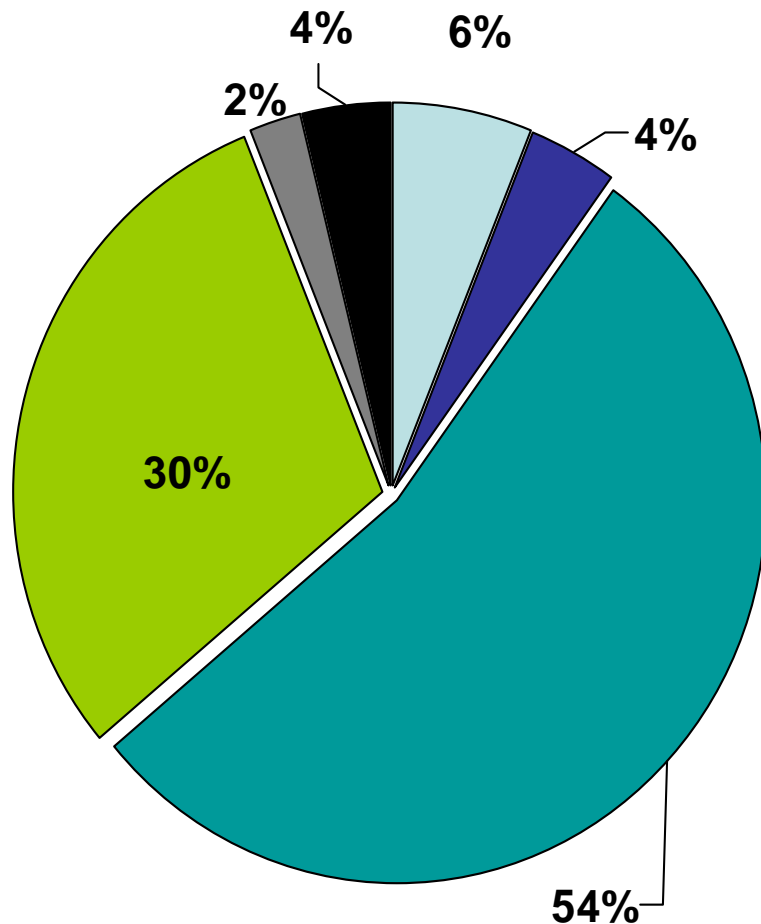


Use Tivoli And zManager To Quickly Create zEnterprise Runtimes For Compile And Test



- Create new VMs for development and test on zEnterprise platforms
- Developers bring up test environments as needed
- Avoid the need for a System z dedicated to test

Rational Tools Improve Software Development For Many Roles And Tasks

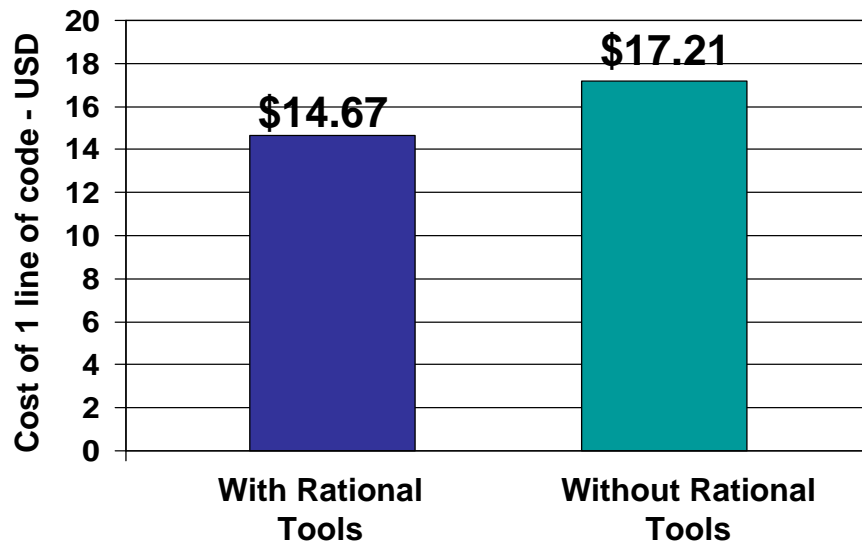


Typical predominance of role in Application Development

	Tasks	Tools
Manager	<ul style="list-style-type: none"> Work with stakeholders on requirements Collaborate with team "Own" product releases, track activities Assign work items from requirements 	<ul style="list-style-type: none"> RRC RTC RQM
Architect	<ul style="list-style-type: none"> Map requirements to designs Model application architecture Coordinate models to development Collaborate with team 	<ul style="list-style-type: none"> RRC RTC RSA
Developer	<ul style="list-style-type: none"> Review designs Write code Unit test Collaborate with team 	<ul style="list-style-type: none"> RRC RTC RAD RDz RDp
Tester	<ul style="list-style-type: none"> Design tests Execute tests Report defects Collaborate with team 	<ul style="list-style-type: none"> RRC RTC RQM RFT/RPT
Bld Eng.	<ul style="list-style-type: none"> Nightly builds Test and production builds Deployments Collaborate with team 	<ul style="list-style-type: none"> RTC
Document	<ul style="list-style-type: none"> Monitor requirements Coordinate with developers Write documentation Collaborate with team 	<ul style="list-style-type: none"> RRC RTC

Development Labor Model

- We need a labor model to assess the productivity benefits of Rational tools for System z
- The labor model is created by best fit to customer, industry and IBM internal productivity studies
- Productivity benefits result in labor savings that outweigh the tool costs
- The labor model demonstrates that using Rational tools results in 15% overall development cost savings compared to ISPF



Based on

- 6 Lines of quality code per hour with participation by all team members
- Typical team role distribution
- Productivity improvements from the use of each tool
- Industry average labor, software, infrastructure costs

What IBM Rational Customers Say About ALM Support Built On By Rational Team Concert



“will reduce project overrun costs by 20%.“

--Matt Pomroy - Executive, Software Engineering, Ascendant Technology



"project management dashboards are transparent to everyone"

--Han Jie - Senior Consultant, Siemens



“our developers are more efficient because they are better able to focus on important issues”

--Mika Koivuluoma - Production Manager, TietoEnator



“provides the team transparency and visibility needed to keep work progressing”

--Carson Holmes - Unified ALM Services Manager, Noblestar



“30% productivity gain on our global projects”

--Alain Bergeron - VP Consulting Services, CGI



Thank You

... for coming today

Please remember to fill out
the feedback forms