



# **The New zEnterprise – A Smarter System For A Smart Planet**

Unify Mainframe And Distributed  
Development

# Fit For Purpose Strategy For zEnterprise Requires Common Development Tools

**zEnterprise is a great platform but ... my distributed developers don't talk to my mainframe developers and they use different tools**



**Development Manager**

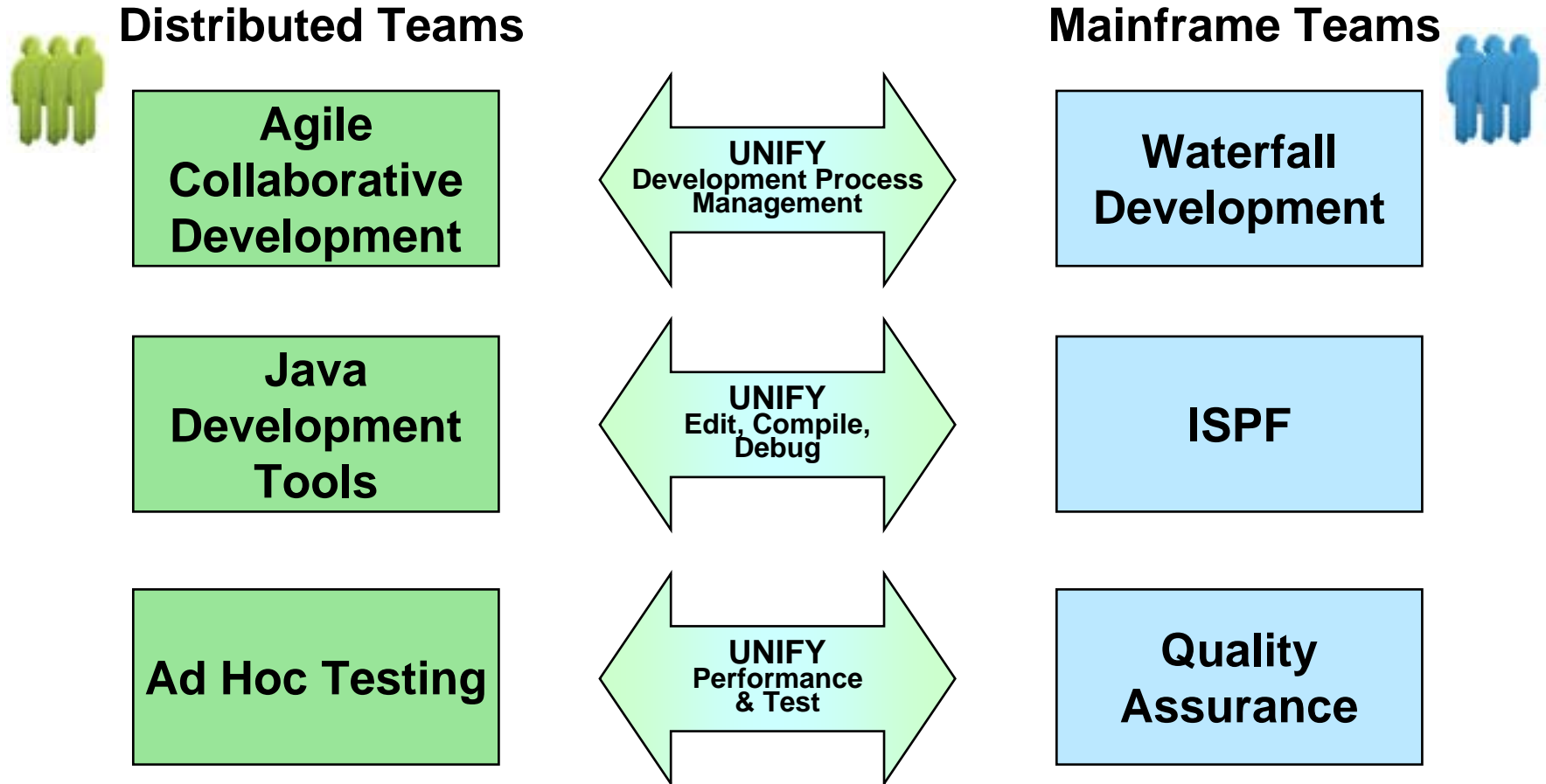
**A unified development environment will get everyone on the same page. . .**

**And improve collaboration and skills transfer**



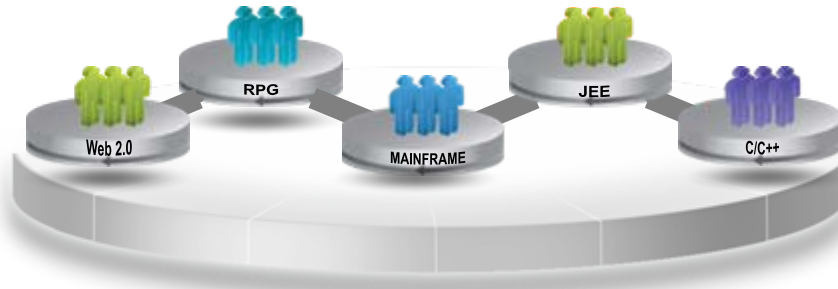
**IBM**

# zEnterprise Combines Mainframe And Distributed Environments In One Platform

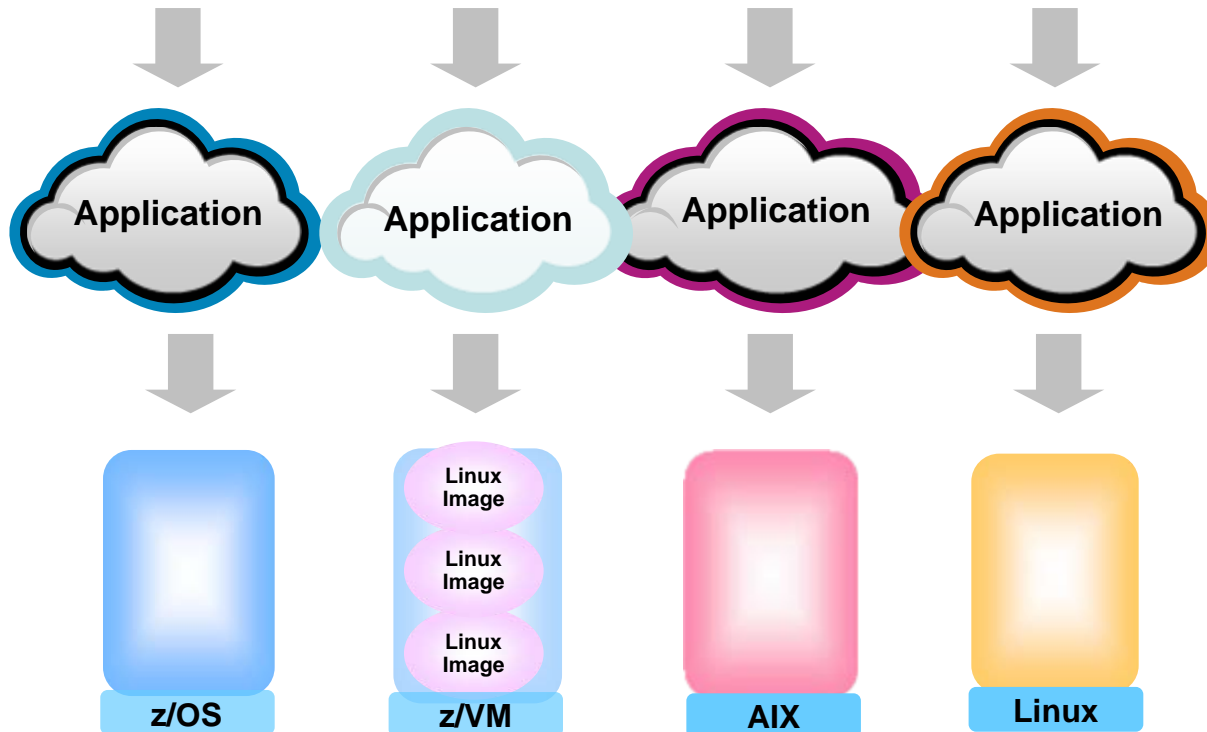


Common processes and shared tools that look and behave the same way for all platforms

# Rational Delivers A Unified Development Tool Set For All These Environments



Unified development processes  
Unified edit, compile, debug  
Unified test



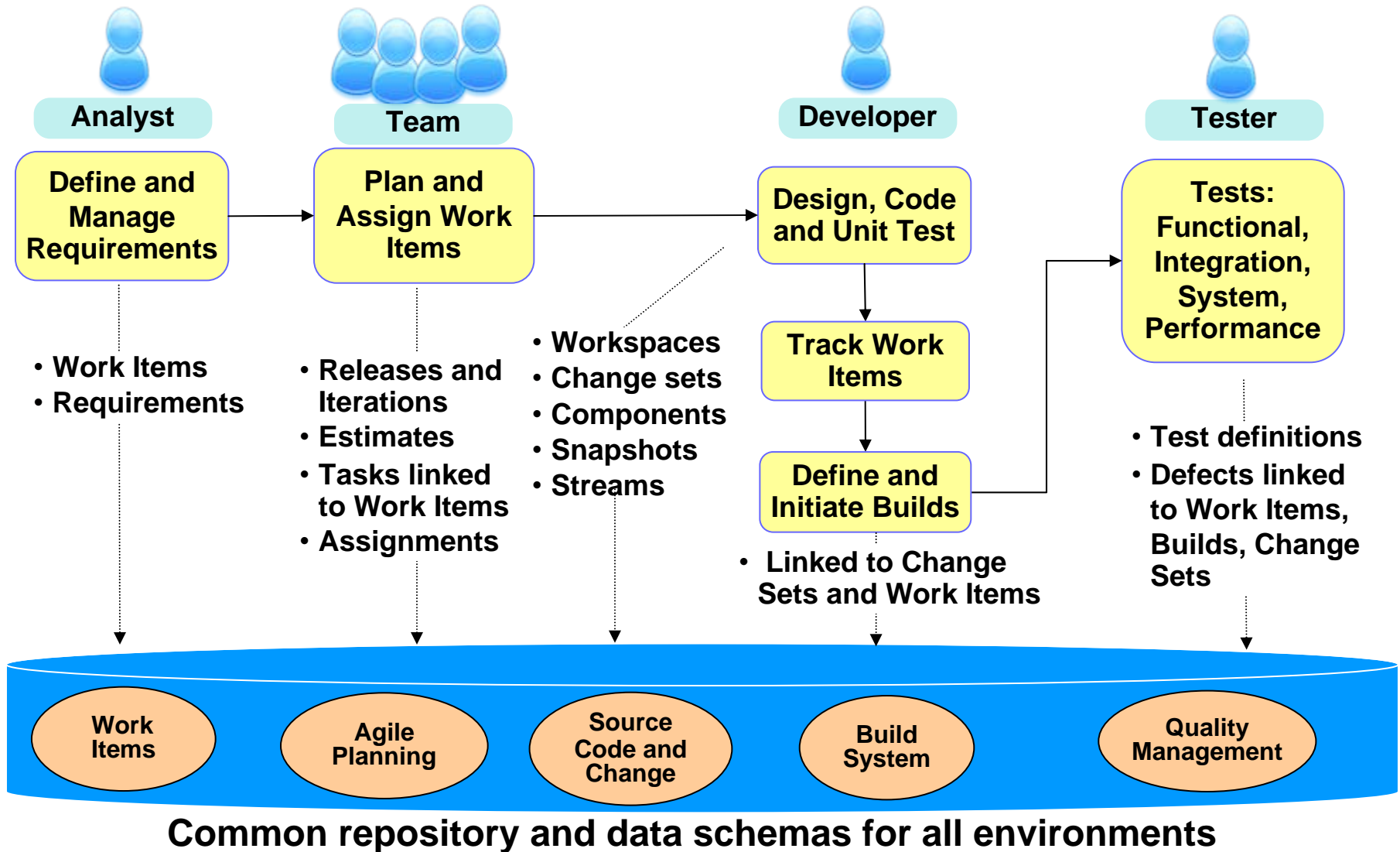
Best fit  
deployment



# Rational Unified Development Tools For zEnterprise

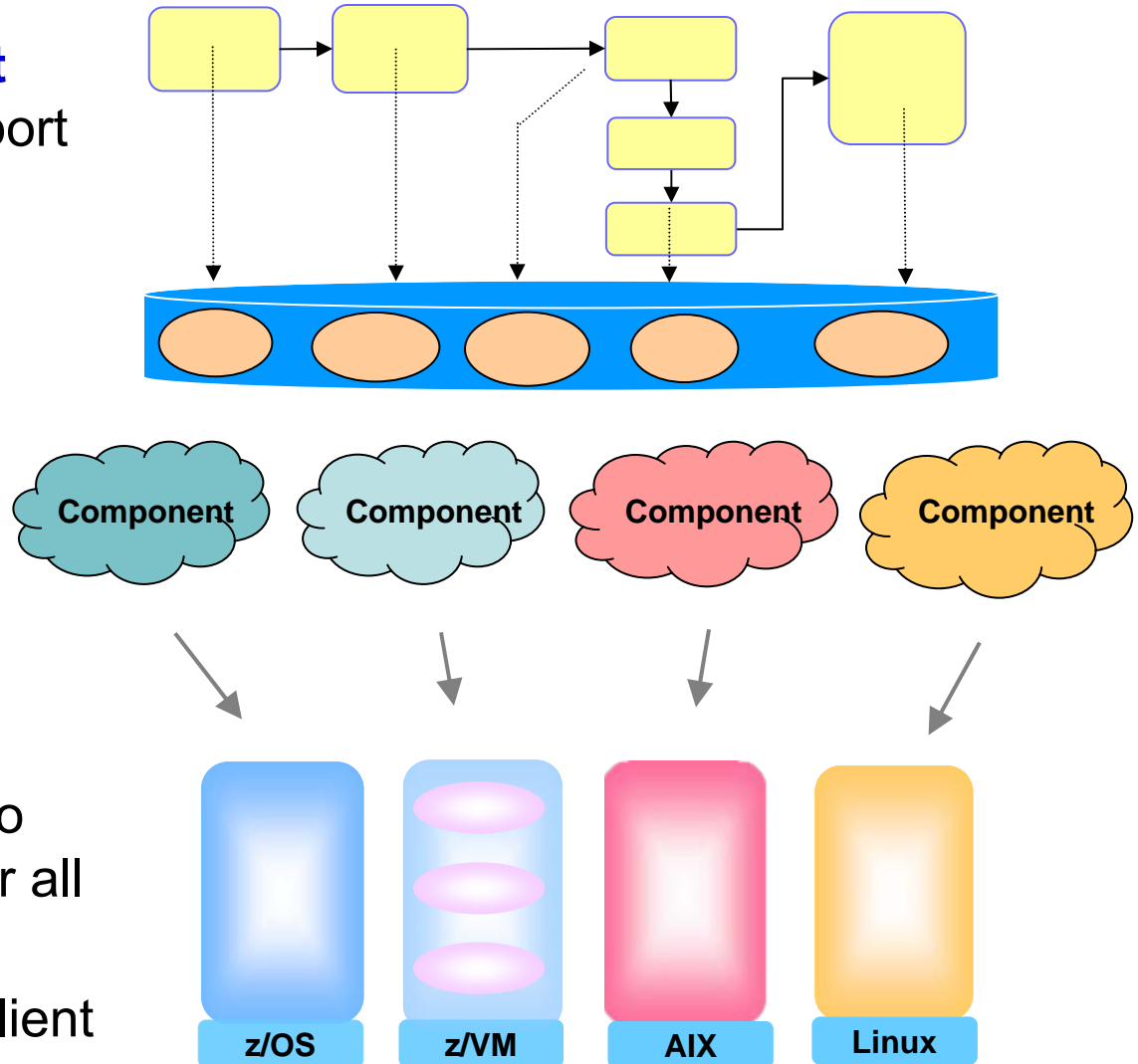
- **Consistent processes for change and release management**
  - ▶ Rational Team Concert For System z
  - ▶ Same process for each component in a hybrid application
- **Consistent edit/compile/debug tools**
  - ▶ Rational Eclipse plug-ins for all environments
  - ▶ IBM compilers optimized for System z and Power environments
  - ▶ Reduce cost, enhance productivity, and encourage skills transfer
- **Common test tools**
  - ▶ Rational Functional Tester and Performance Tester
  - ▶ Rational Quality Manager
  - ▶ Enable test case reuse and skills consolidation
- **Special features to enhance development for z/OS and Linux on System z**
  - ▶ Rational Developer For System z Unit Test
  - ▶ System z Solution Edition for Application Development (z/OS)
  - ▶ Enterprise Linux Server (Linux on System z)
  - ▶ Reduce cost of development for z196 environments

# Rational Team Concert Supports Agile Development Processes

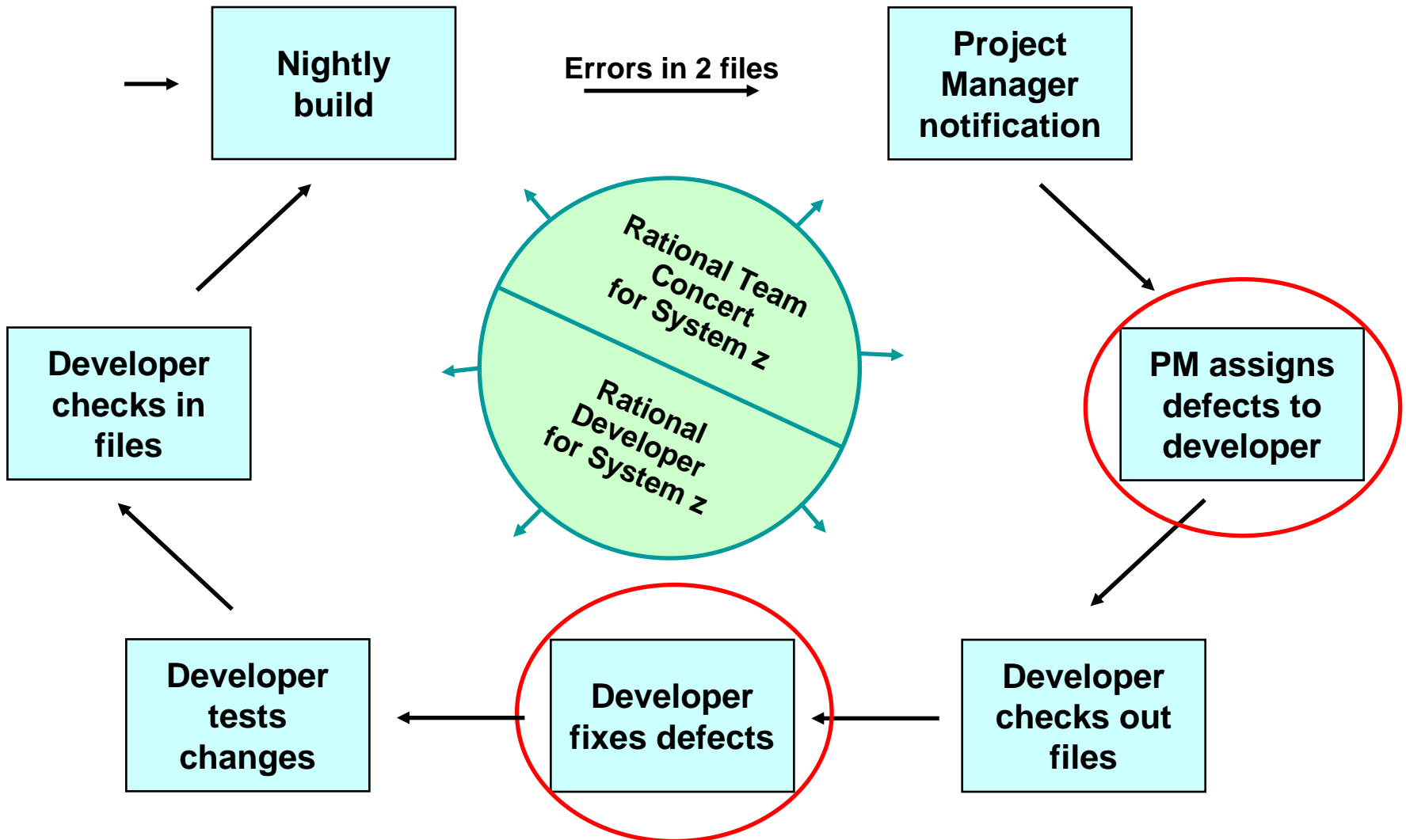


# Unified Development Process Management For All Your Development Teams

- **Rational Team Concert for System z** team support
  - ▶ Team Processes
  - ▶ Project tracking
  - ▶ Collaboration
  - ▶ Repository
- Same tool for
  - ▶ z/OS
  - ▶ Linux
  - ▶ AIX
- A single User Interface to manage development for all environments
  - ▶ Browser or Eclipse client



# Demonstration Scenario





# DEMO: Rational Team Concert For System z

1. In this demo the Project Manager will create a new Work Item and assign it to a Developer
2. In the next demo the Developer will find the newly assigned Work Item and complete it

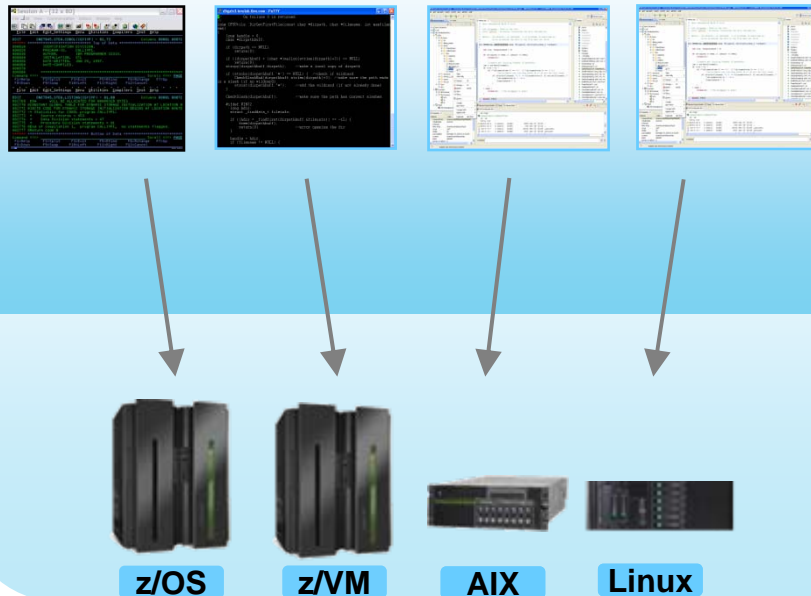
The screenshot displays the Rational Team Concert (RTC) interface for a project named 'Team 1's Plan - M1'. The interface is divided into several sections:

- Left Panel:** A tree view showing the project structure, including 'Repository Connections', 'SOF Modernization Project', 'Builds', 'Data Set Definitions', 'Language Definitions', 'Plans', 'Recently Viewed Plans', 'My Current Plans', 'SOF Dev Plan 1 [M1]', 'Team 1's Plan [M1]', 'Current Plans', 'All Plans', 'Reports', 'Source Control', 'Work Items', 'Debug', 'Favorites', 'Feeds', 'My Repository Workspaces', 'My Team Areas', and 'Work Item History'.
- Center Panel:** A list of team members and their assigned work items. A blue circle highlights the work item 'fix defects in two files for Natick' assigned to Haze. Other work items include 'Define a new build' (Avijit), 'Share code with Jazz Source Control' (Fehmina), 'Define permissions' (Jess), and 'Discuss feature list with stakeholders' (Li).
- Right Panel:** A 'View As' section with options like 'Developer's Taskboard', 'Planned Time', 'Ranked List', 'Team Folders', and 'Work Breakdown'. Below it are 'Actions' (Edit | Copy, Re-sort) and 'Exclude' options (Assigned Items, Empty Groups, Estimated Items).
- Bottom Panel:** A table showing a list of work items. The table has columns for 'I.', 'Status', 'P', 'S', 'Summary', 'Owned By', and 'Created By'. The work items listed are:

I.	Status	P	S	Summary	Owned By	Created By
22	New			fix defects in two files for Natick	Haze	Jess
11	New			Discuss feature list with stakeholders	Li	Jess
10	New			Fix text on splash screen	Monk	Jess
7	New			Define permissions	Jess	ihm...

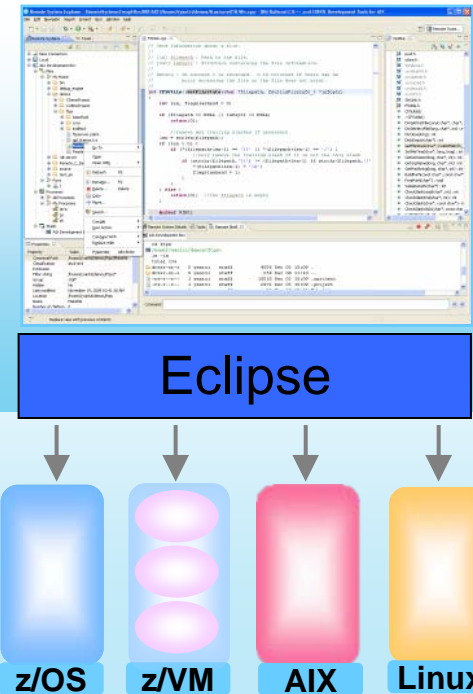
# Rational Developer Tools For zEnterprise

## Before...



- Separate tools for each platform
- Green screen/command-line editors
- GUI editors

## Rational Developer Tools

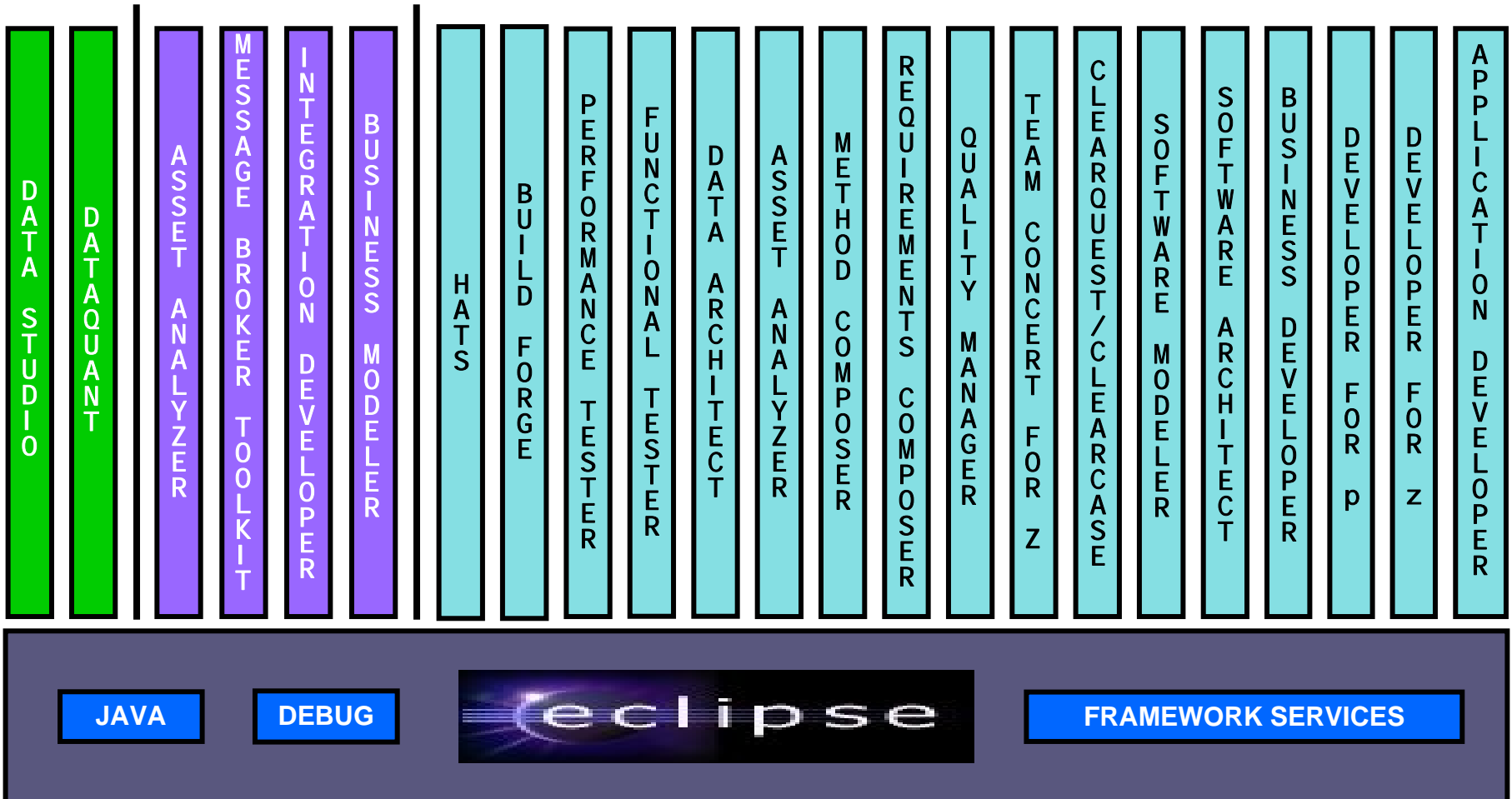


- Eclipse-based IDEs with modern GUI
- Plug-ins support all environments
- 30% or more improvement in developer productivity<sup>1</sup>

<sup>1</sup>Based on IBM customer study, ["Making a Business Case for IBM Rational Developer for z"](#)

# Eclipse Plug In Framework

- Rational Edit, Compile, and Debug tools are built on Eclipse
- Open Source Eclipse framework enforces consistent tool behavior and consistent user interfaces



# Rational Developer For zEnterprise Family Of Products

The IBM Rational Developer Family of products includes integrated development tools for the major development workloads on z/OS, Linux on System z, AIX, and Linux

## ■ Rational Application Developer (RAD)

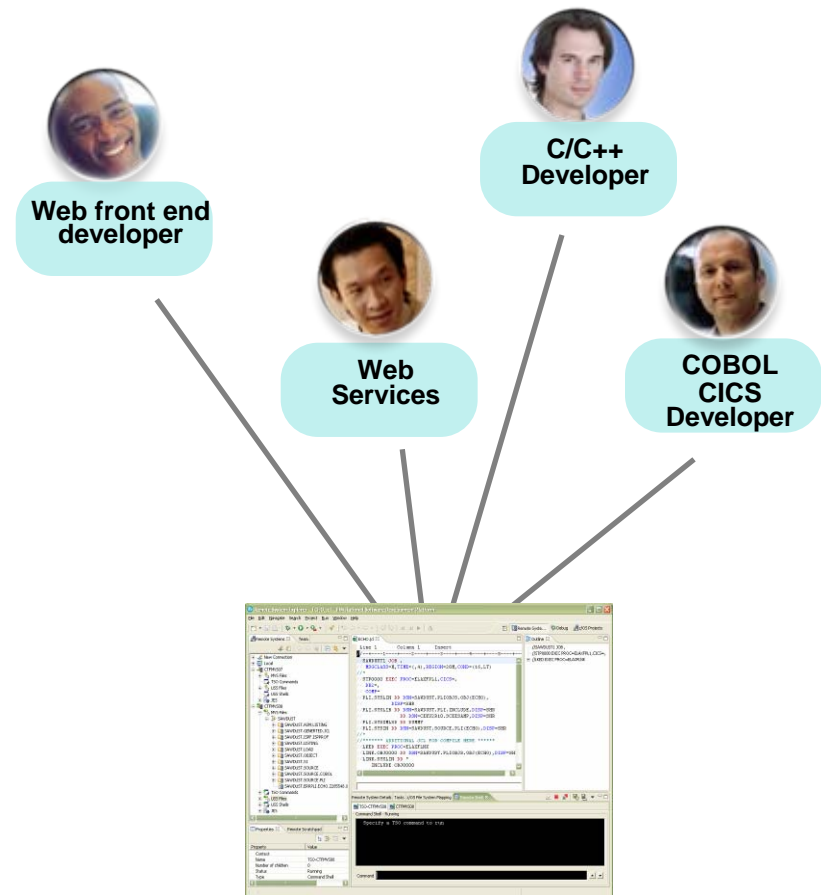
- ▶ Develop Java, JEE, and JavaScript
- ▶ Tools for Web, Web Services, and Web 2.0 designers and developers
- ▶ WebSphere Application Server Test Environment

## ■ Rational Developer for System z (RDz)

- ▶ Develop traditional C/C++, COBOL, HLASM, CICS, and IMS applications
- ▶ Tools for Web, Web Services, and SOA designers and developers
- ▶ Provides remote file system access and includes C Developer Tools (CDT) for Linux
- ▶ Can include RAD

## ■ Rational Developer for Power Systems Software (RDp)

- ▶ C/C++, COBOL and Fortran development tools for AIX
- ▶ Can include RAD



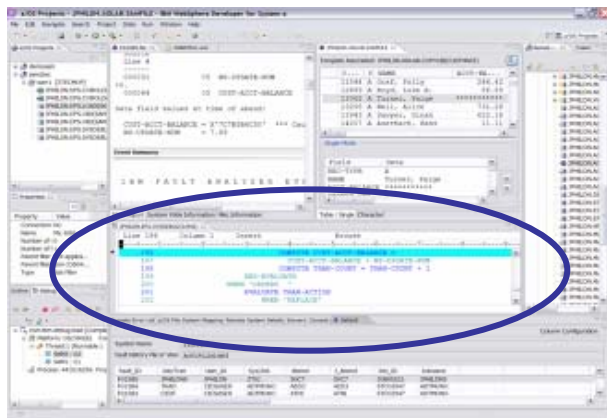
# Rational Developer For zEnterprise Family Of Products

Plug-in:	RAD 7.6	RDz 7.6	RDp 7.6
Based on Eclipse 3.4	X	X	X
Installation Manager install and updates	X	X	X
Cheatsheets, tutorials, samples, content assist	X	X	X
Java development tools (JDT) - non-JEE	X	X	X
JEE development tools	X	X	X
JEE Web development tools	X	X	X
WAS test environment	X	X	X
Portal development tools	X	X	X
Data tools (database connectivity, SQL, etc.)	X	X	X
WAS Feature Pack for Web 2.0 (Dojo, ReST adapter for JEE apps)	X	X	X
Rational Team Concert client	X	X	X
XML development tools	X	X	X
Code Quality, Testing and Deployment tools		X	X
EGL tools: Web 2.0 Rich UI, Java, Java Web application (AIX), COBOL (zNext) generators		X	X
Editors/debuggers for C/C++ (z/OS), COBOL (z/OS), PL/I, HLASM, JCL		X	
Eclipse CDT editor/debugger for C/C++ (Linux on System z, Linux)		X	
Editors/debuggers for C/C++ (AIX), COBOL (AIX), FORTRAN (AIX), Assembler development tools			X

# Debug zEnterprise Applications From The Workstation

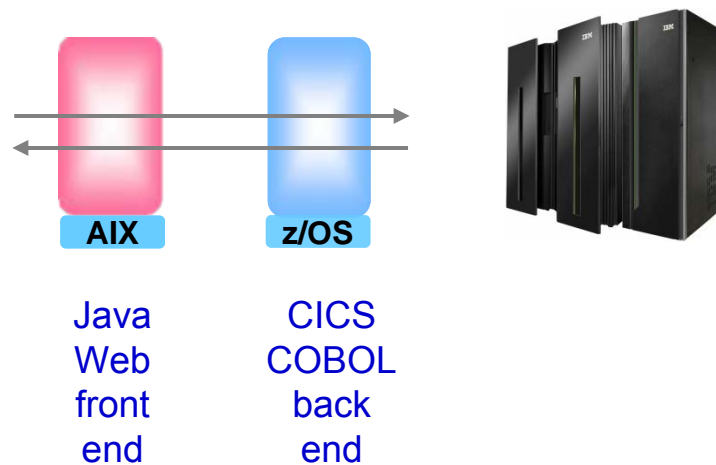
- Debuggers built on the common Eclipse framework enable
  - ▶ Debug step across languages
  - ▶ Debug step across environments
- Collaborative debugging when integrated with RTC

Work with code in the debugger here



Workstation-based  
RAD, RDz, RDp

Debug applications running here



Java  
Web  
front  
end

CICS  
COBOL  
back  
end

# Edit, Compile And Debug With ISPF

Wait an indeterminate time for the job queue and possible failed completion

Start here

submit compile job

swap to SDSF

select job

find error msg

find code line  
(remember error)

swap to edit session

exit JCL

edit JCL

exit source

change code

find code line

edit source

```
Session A - [32 x 80]
File Edit View Communication Actions Window Help
File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT DNET045.STEW.COBOL(IGYIYP) - 01.73 Columns 00001 00072
***** Top of Data *****
000010 IDENTIFICATION DIVISION.
000020 PROGRAM-ID. CALLIYP1.
000030 AUTHOR. IBM PROGRAMMER 33333.
000040 INSTALLATION. STL
000050 DATE-WRITTEN. JAN 25, 1997.
000060 DATE-COMPILED.
000070
000080
Command ==> Scroll ==> PAGE
F1=Help F2=Split F3=Exit F5=RFind F6=Rchange F7=Up
F8=Down F9=Swap F10=Left F11=Right F12=Cancel
File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT DNET045.STEW.LISTING(IGYIYP) - 01.00 Columns 00001 00072
002769 DSA WILL BE ALLOCATED FOR 00000320 BYTES
002770 0CONSTANT GLOBAL TABLE FOR DYNAMIC STORAGE INITIALIZATION AT LOCATION 00
002771 0INIT0 CODE FOR DYNAMIC STORAGE INITIALIZATION BEGINS AT LOCATION 006700
002772 +- Statistics for COBOL program CALLIYP1:
002773 + Source records = 453
002774 + Data Division statements = 47
002775 + Procedure Division statements = 91
002776 0End of compilation 1, program CALLIYP1, no statements flagged.
002777 0Return code 0
***** Bottom of Data *****
Command ==> Scroll ==> PAGE
F1=Help F2=Split F3=Exit F5=RFind F6=Rchange F7=Up
F8=Down F9=Swap F10=Left F11=Right F12=Cancel
38/01
```

- Programmer goes through a sequence of screens in order to get the job done
  - ▶ ISPF 3.4 listings, job listings, SDSF outputs, etc.
- Programmer is constantly flipping back and forth between these ISPF screens
  - ▶ Easy for experienced mainframe programmers but not for newbies

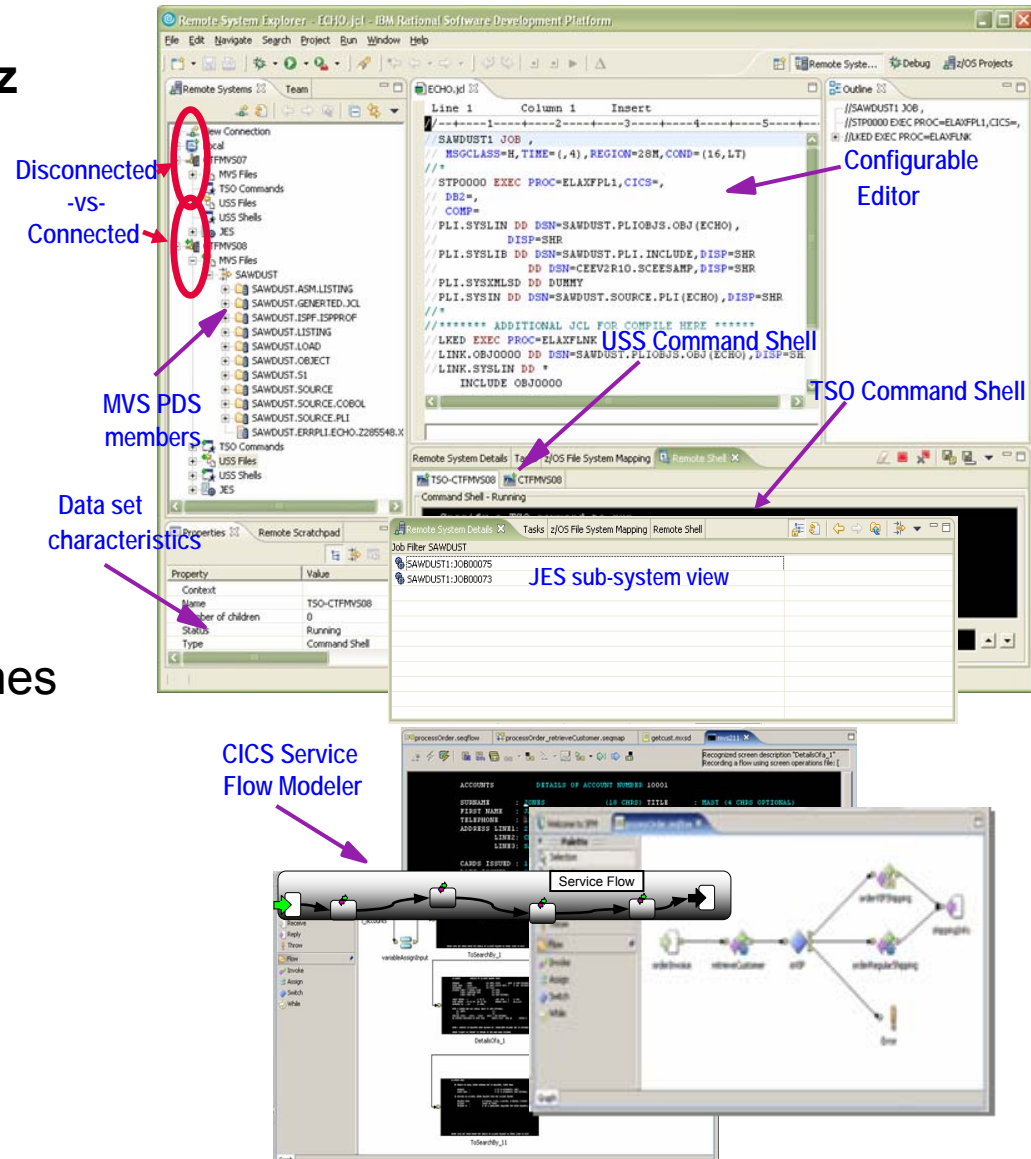
# Edit, Compile And Debug With Rational Developer For System z

## ■ Modern development for System z

- ▶ Works like other IBM tools for distributed platforms

## ■ RDz supports development and reuse of Enterprise assets

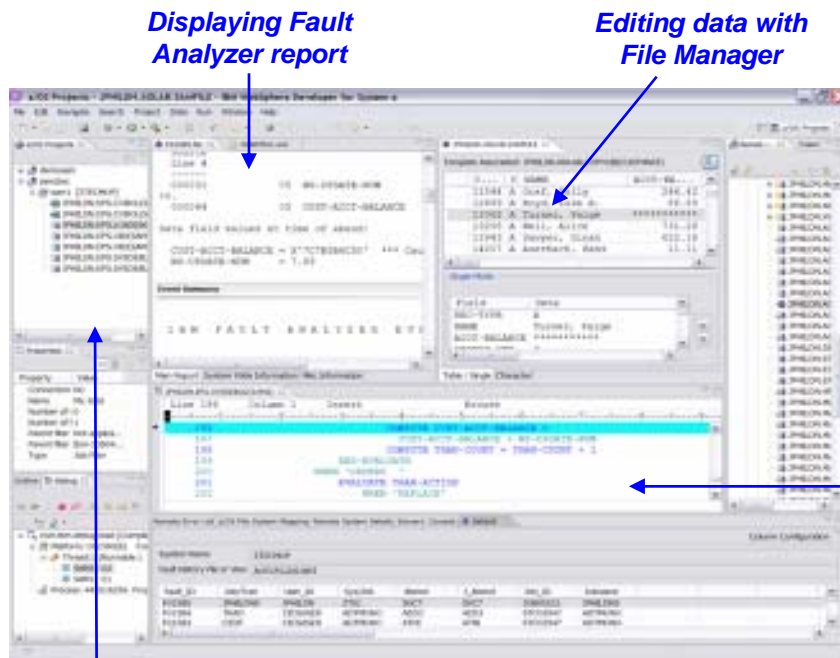
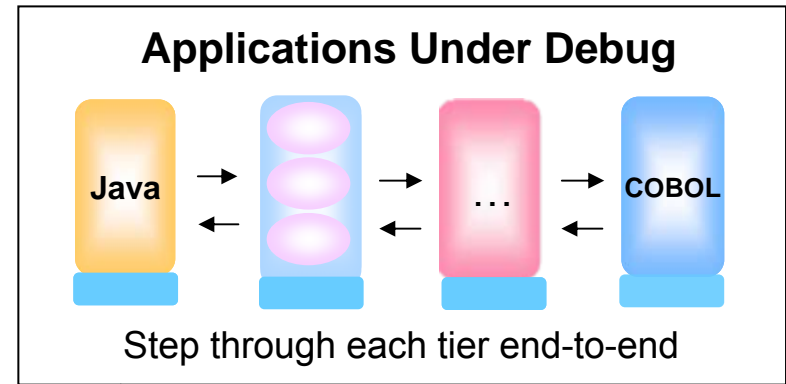
- ▶ Support for COBOL, PL/I, C, C++, HLASM, Java, EGL and Web services
- ▶ Supports existing and new runtimes
  - CICS, IMS, Batch, USS, DB2, WAS
- ▶ Interactive access to z/OS for
  - Development, debug, job generation, submission, monitoring, command execution





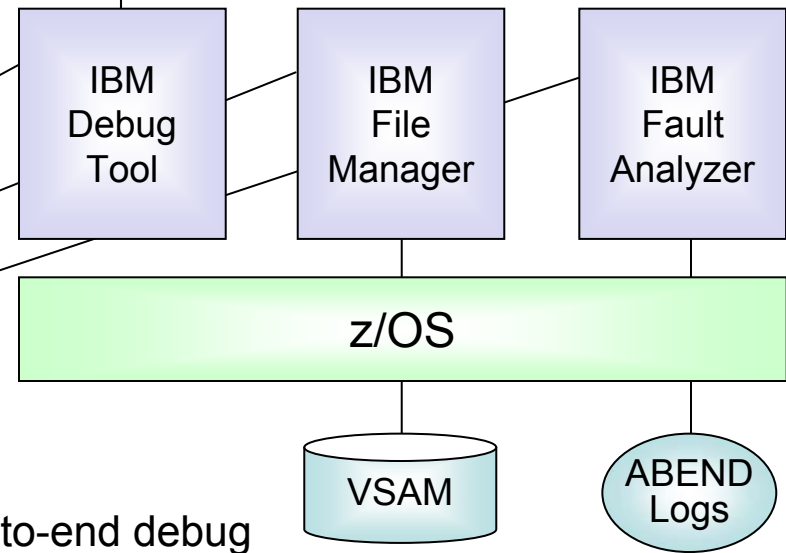
# Interactive Problem Determination: RDz Integrates With Host-based Tools

- Work with the PD Tools through the RDz client
- Easy access to all three tools at the same time
- Debug and step through multi-tier applications
  - ▶ Distributed *and* mainframe
  - ▶ Same debugger as for distributed systems



*Developing System z application with RDz*

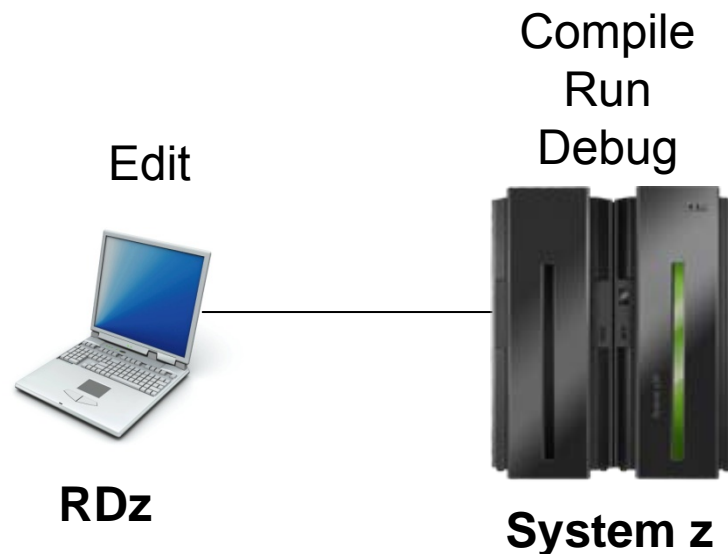
## RDz Workstation



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

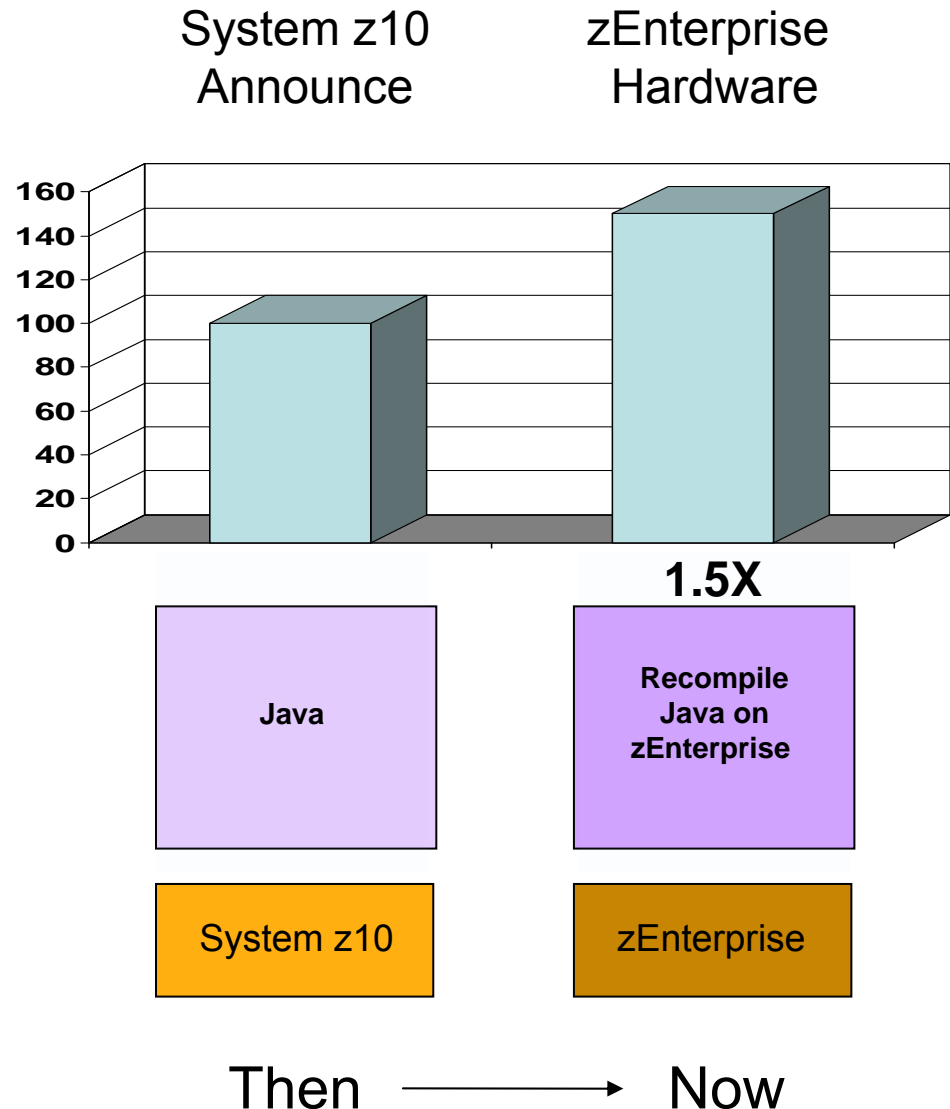
# DEMO: Work With COBOL And Java Using Rational Developer For System z

- Perspectives and views
- Remote System Explorer
- Working with host datasets
- Productivity features
  
- Based on the assigned work item, the Developer will fix the problem code using the language-sensitive COBOL and Java editors



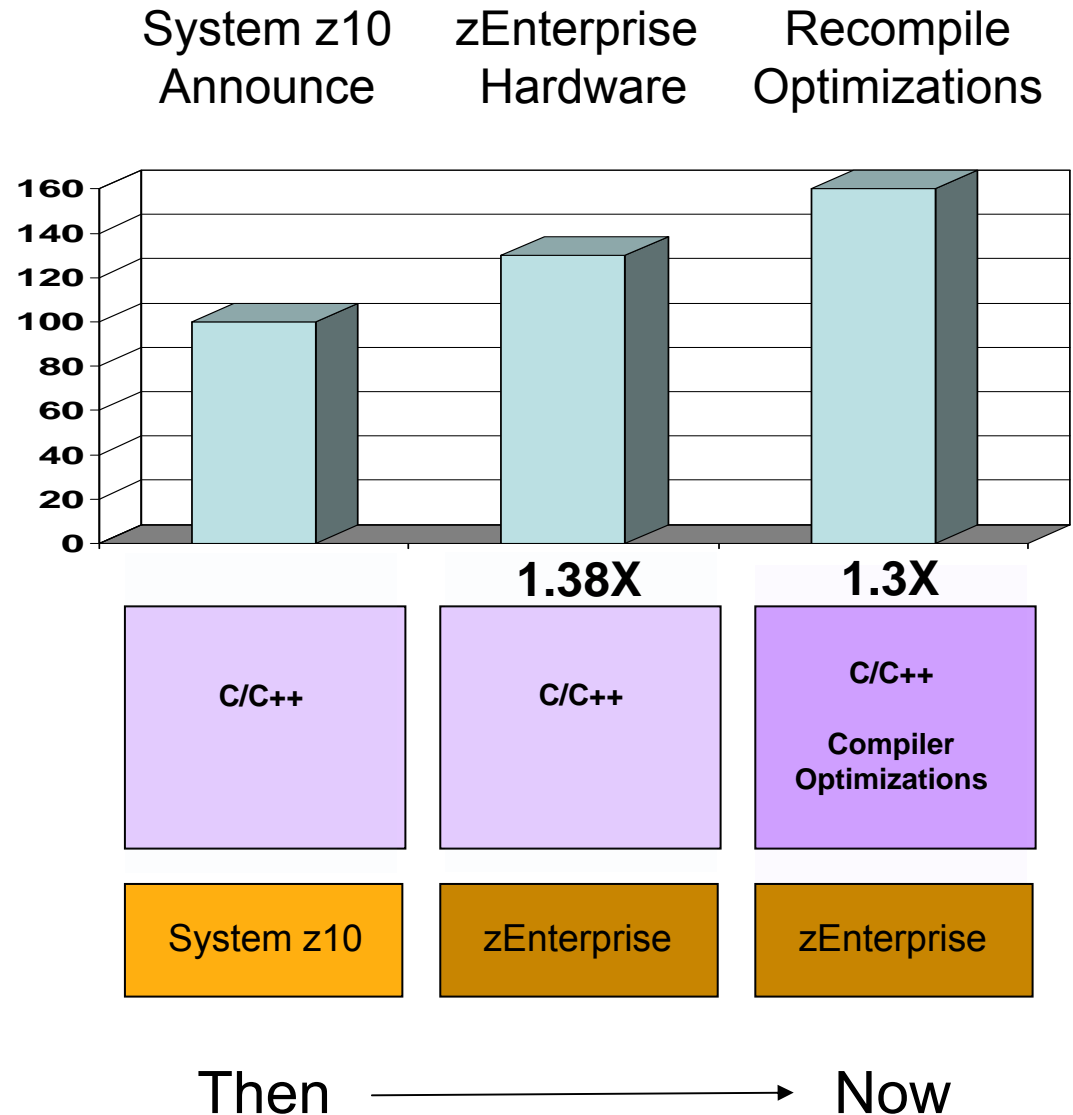
# Continued Java Optimizations For z/OS – From Then To Now

- Continued investment to optimize Java software for z/OS environment
  - ▶ Uplevel to zEnterprise hardware and recompile Java produces an average of 1.5 times performance improvement
  - ▶ Per thread performance with Java6SR8
    - ▶ CPU-benchmark +63%
    - ▶ Multi-threaded +45%
    - ▶ ILOG/CONFIRM +45%



# Continued C/C++ Compiler Optimizations For z/OS – From Then To Now

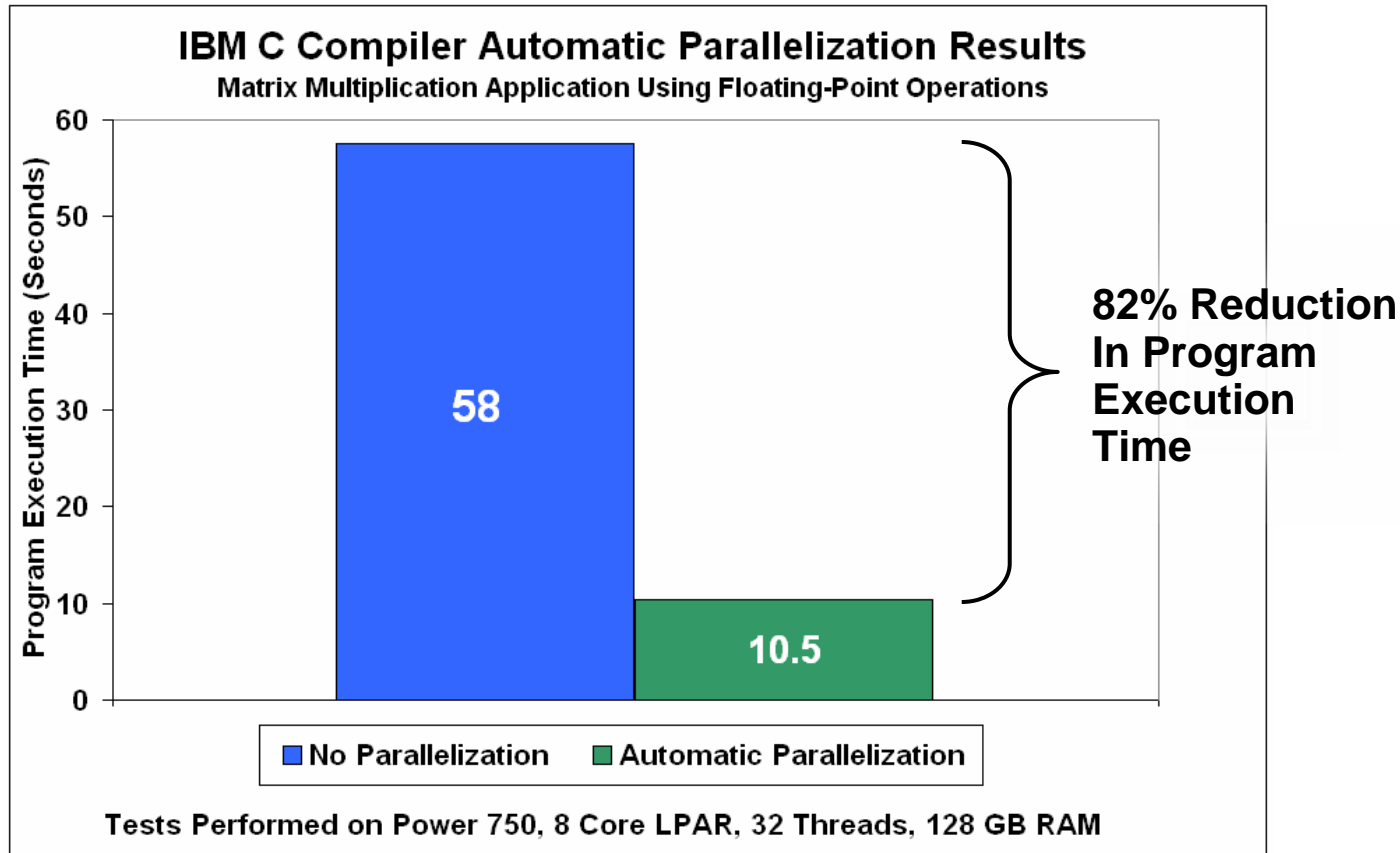
- Continued investment to optimize key software for z/OS environment
  - ▶ Uplevel to zEnterprise hardware produces 1.38 times performance improvement
  - ▶ Recompile C/C++ using compiler optimization produces 1.3 times performance improvement
  - ▶ From then to now – almost **1.8** times performance improvement



# C/C++ Compiler Optimizations For Power Blade On zEnterprise

- **Optimizations** for improving performance (no code changes needed)
  - ▶ Elimination of redundant code
  - ▶ Loop optimization
    - Better loop scheduling
    - High-order loop analysis and transformations
  - ▶ Vectorizes calls to system math functions by calling the equivalent MASS vector functions
  - ▶ Elimination of compile-time memory usage limits
  - ▶ Reorganization or elimination of global data structures
  
- Parallelization improves performance on multi-core systems
  - ▶ **Automatic parallelization** of iterative program loops (eg. do, while, for, etc.)
    - Iterations are executed concurrently on all available processor cores
    - No code changes needed
  - ▶ **Explicit parallelization** using the OpenMP Application Program Interface Version 3.0 specification
    - Must add API calls

# IBM Compiler Automatic Parallelization Improves Scientific Workload Performance On Multi-Core Power Systems Without Code Changes



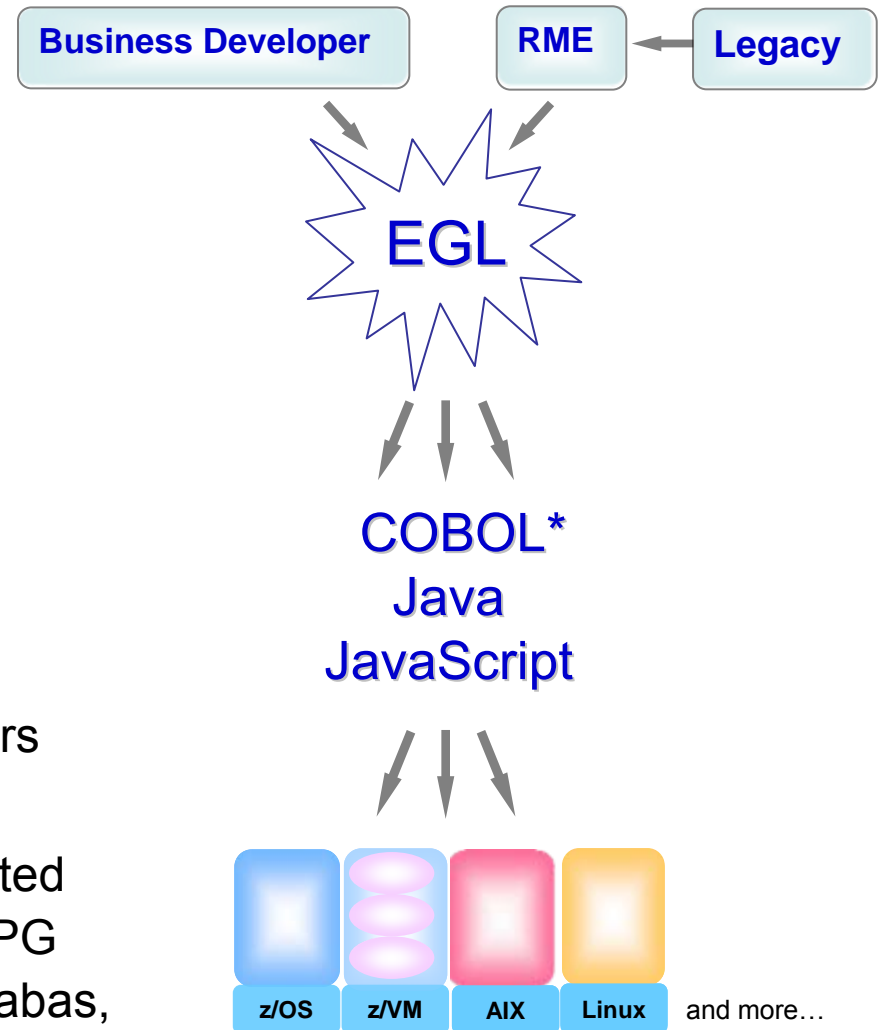
- Automatic parallelization of iterative program loops
- Iterations are executed concurrently on all available processor cores and threads
- No code changes needed

Source: Internal CPO Study

# EGL – One Language Fit for Every Purpose

## Enterprise Generation Language

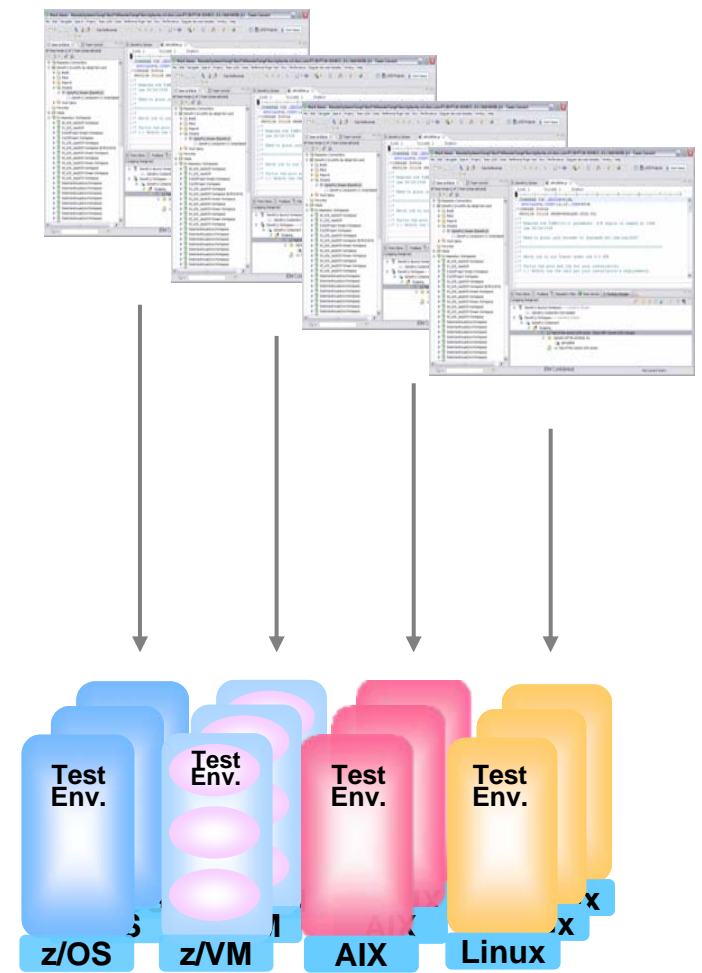
- 4<sup>th</sup> generation business language
- Deploy everywhere based on best fit for purpose
- **Rational Business Developer (RBD)**  
And RDz and RDp
  - ▶ EGL development tools for System z and Power Systems
  - ▶ For Business Developers
  - ▶ Tools for Web, Web Services, & Web 2.0 designers and developers
- **Rational Migration Extension (RME)**  
Provides on average over 95% automated migration to EGL from legacy Synon RPG and COBOL, CA Cool:Gen, Natural Adabas, and green screens



\* COBOL on z/OS, (IBM i)

# zEnterprise Virtualization Quickly Provides Runtimes For Compile And Test In All Environments

- Virtual machines provisioned for compile/test in each environment
- Special lower cost offerings:
  - ▶ System z Solution Edition for Application Development
    - LPAR-based addition of a customized package of hardware, compiler, middleware, and maintenance for 3 years
    - For compile, unit and system test with z/OS
  - ▶ Solution Edition for Enterprise Linux
    - LPAR-based addition of hardware, z/VM, and maintenance for 3 years
    - Can be used for compile, unit and system test with Linux on System z





# Unit Test Option For z/OS Applications On The Workstation



## Rational Developer for System z Unit Test (RDz UT)

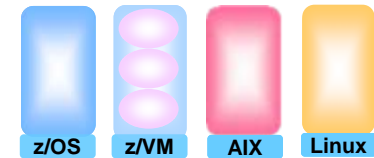
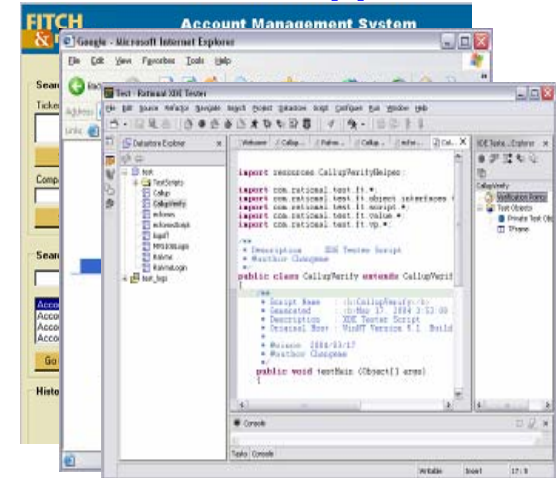
- z/OS runtime environment runs on x86 Linux workstation
  - ▶ **Compile and unit test** on the workstation - no zEnterprise hardware needed
  - ▶ Emulates System z general purpose processors, zIIPs, and zAAPs
- Reduce development MIPS for z/OS applications
  - ▶ Lower cost and better productivity
  - ▶ Enable new skills quickly
- Includes compilers, middleware, RDz & RTCz server load modules
- USD \$4K / user license (in addition to \$5,670 for RDz)



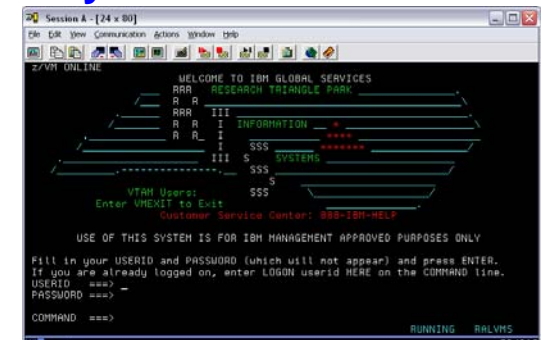
# Unified Testing with Rational Functional Tester And Rational Performance Tester

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

## Web and GUI Applications

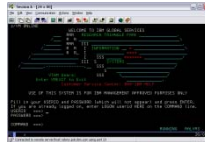


## System z Terminal UI



# Compare Development Costs Over Three Years

## CASE 1



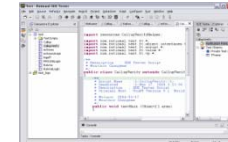
20 programmer team  
@ \$140K/yr/person

ISPF for Edit, Compile, Debug

Use production MIPS

- Normal programmer productivity
- Consume MIPS for edit, compile, debug and test
- High cost production MIPS

## CASE 2



20 programmer team  
@ \$140K/yr/person

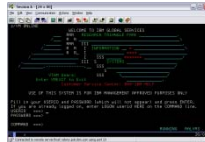
RDz for Edit, Compile, Debug

System z Solution Edition for  
Application Development

- 30% better programmer productivity
- No MIPS consumed for edit
- Significantly lower cost MIPS for compile, debug and test

# Compare Development Costs Over Three Years

## CASE 1



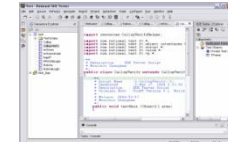
20 programmer team  
@ \$140K/yr/person

ISPF for Edit, Compile, Debug

Use production MIPS

- Programmer cost \$8.4M
- MIPS cost \$4.5M
- Total cost \$12.9M

## CASE 2



20 programmer team  
@ \$140K/yr/person

RDz for Edit, Compile, Debug

System z Solution Edition for  
Application Development

- Programmer cost \$6.7M
- Net MIPS cost \$0.2M
- Total cost 47% less \$6.9M

# Benefits Of Rational Common Development Tools

- Unify development teams via common development tools for zEnterprise
- Enhance cross-platform development skills
  - ▶ Easier for Java developers to learn COBOL skills
  - ▶ Easier for COBOL developers to learn Java skills
- Structured common processes, tools, and special offerings can reduce development costs by up to 47%