Application Development for z/OS

Not your Father's Green Screen

Mike Fulton, IBM Distinguished Engineer

CTO DevOps for Enterprise Systems

Thursday, August 13th, 2015

3:15pm to 4:15pm Dolphin, Oceanic 7

#SHAREorg

CELEBRATING
60
YEARS
OF SHARE
Influencing IT Since 1955

Abstract

Ask most people how they write and maintain applications on z/OS and you hear "oh, you use this thing called a green screen" followed by a chuckle.

In reality, application development for zEnterprise applications has been transformed over the past several years to the point where application developers enjoy the same or better features from integrated development environments as programmers who work on other platforms.

Advances in remote system communication and interaction, syntax-highlighting, parsing, and code understanding for Assembler, PL/I, C/C++, and COBOL source code, as well as programming assists such as code snippets and templates are all available to application programmers. Interactive debug of applications, written in multiple programming languages and running in various runtime environments is also possible and can greatly boost programmer productivity.

Come and learn about how these features can enable application developers who are new to the mainframe to interact with, update, and efficiently enhance mainframe applications.

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
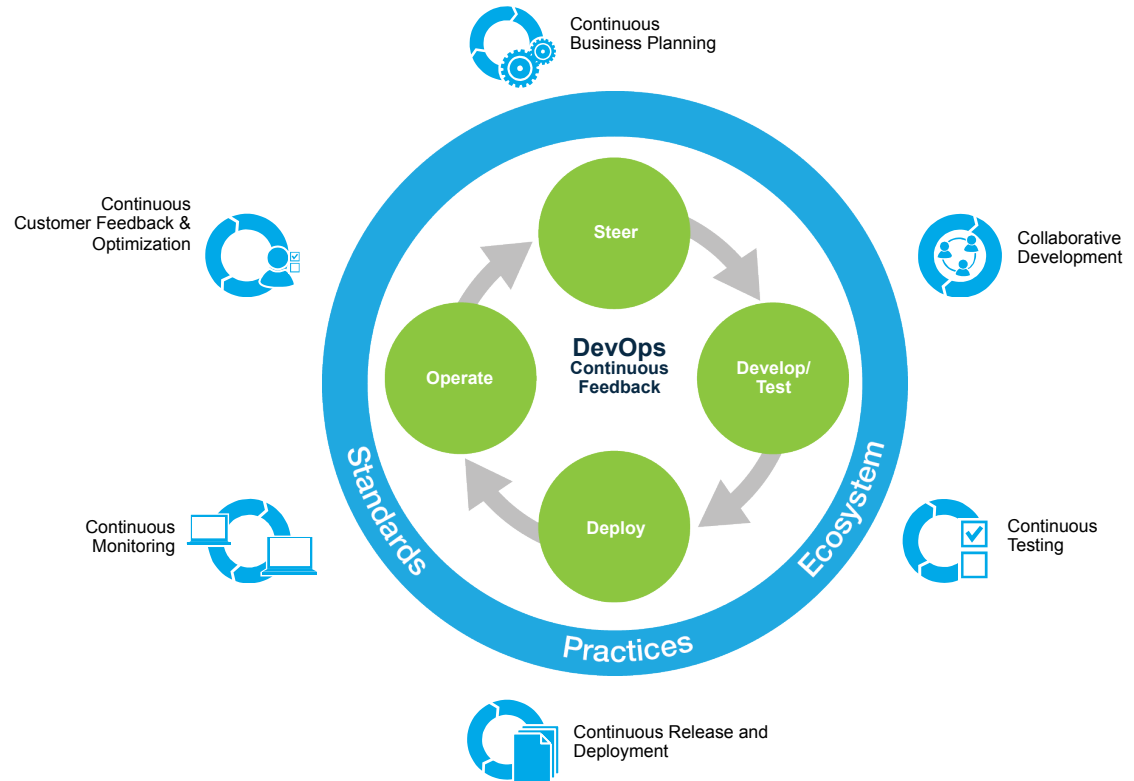
August 2015

IBM DevOps point of view

Enterprise capability for continuous software delivery that enables organizations to seize market opportunities and reduce time to customer feedback

Accelerate software delivery – *for faster time to value*

Reduce time to customer feedback – *for improved customer experience*

Balance speed, cost, quality and risk – *for increased capacity to innovate*

Continuous Business Planning

Collaborative Development

Continuous Testing

Continuous Release and Deployment

Continuous Monitoring

Continuous Customer Feedback & Optimization

Steer

Develop/ Test

Deploy

Operate

DevOps Continuous Feedback

Standards

Ecosystem

Practices

Clients are achieving **measurable business outcomes with DevOps**

SHARE in Orlando 2015

Collaborative Development

Integrated Application Lifecycle Management for all platforms
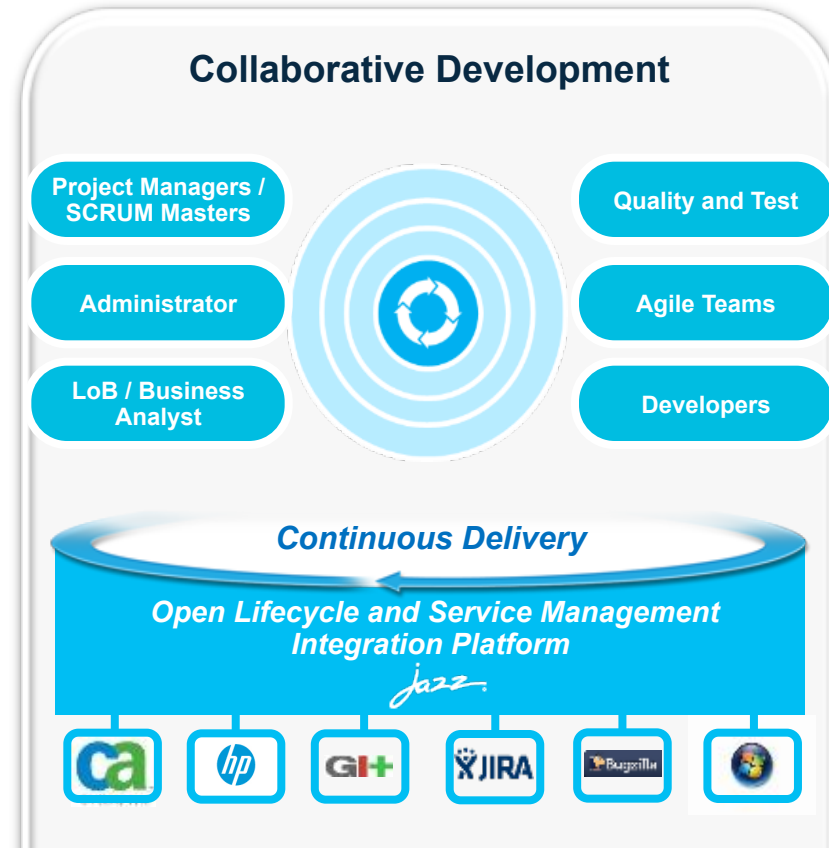
Accelerate **time to delivery** with real-time planning

Improve **quality** with lifecycle traceability

Maximize **time to value** with in-context collaboration

Refine **predictability** with development intelligence

Reduce **costs** with continuous improvement

✓ **Manage many sources - JavaScript to COBOL**
✓ **Single work item can span technologies & solutions**
✓ **Quickens agile adoption on the mainframe**
✓ **Integrates existing SCMs and deployment tools**

IBM Rational Collaborative Lifecycle Management (CLM)
- IBM Rational Team Concert
- IBM Rational Doors Next Generation
- IBM Rational Quality Manager

Collaborative Development

Project Managers / SCRUM Masters

Administrator

LoB / Business Analyst

Quality and Test

Agile Teams

Developers

Continuous Delivery

Open Lifecycle and Service Management Integration Platform

jazz.

SHARE
in Orlando 2015

Rational Team Concert: One tool, many capabilities

Work Items

Planning

Source Control

Builds – Continuous

Dashboards & Reporting
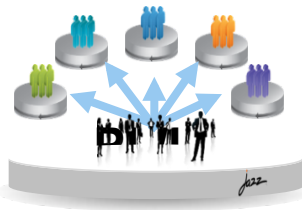

Enforcement and Automation

Rational Developer for System z:
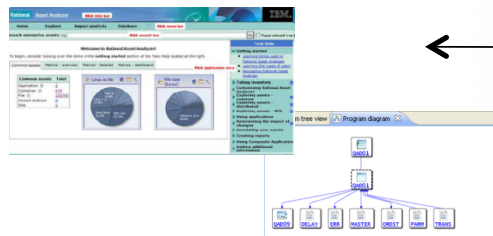## An Integrated Development Environment for System z

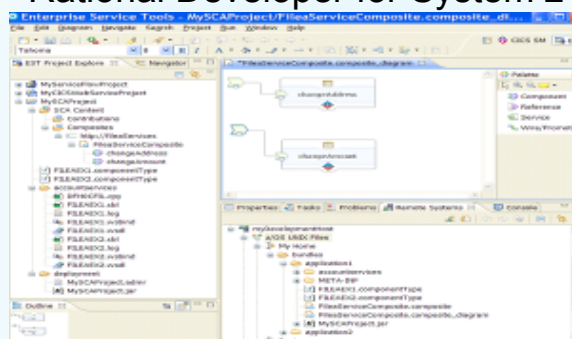Integration with Team Concert for Lifecycle and Source Management

Access to typical System z sub-system functionality in z/OS, CICS, IMS, DB2, WAS
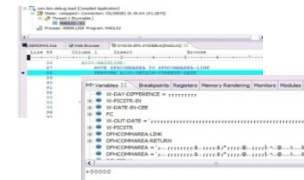
Integration with Asset Analyzer for Application Understanding and Impact Analysis
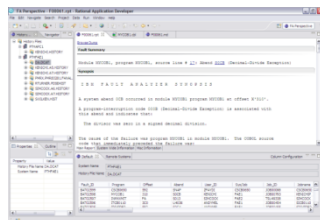
Rational Developer for System z

New

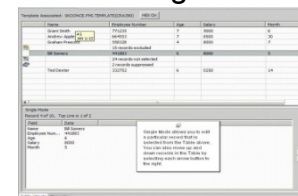**Out of the Box debugger and code coverage capabilities**

A modern IDE for productive development of cross-platform applications written in COBOL, PL/I, ASM, Java, EGL or C/C++ in System z CICS, IMS, DB2, Batch applications

Integration with Fault Analyzer for Dump Analysis

Integration with File Manager for file and test data handling

Integration with RD&T for flexible access to System z environment

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

SHARE in Orlando 2015

August 2015

Rational Developer for System z
RDz: stay focused: work **concurrently not sequentially**

Edit a program

Access Datasets + Dataset Management

File Compare

Submit a Compile | **File Search**

Dataset Statistics

Access Jobs (Outlist facility)

Rational Development and Test Environment for System z
The ultimate in modern application development for System z

COBOL, PL/I, C++, Java, EGL, Batch, Assembler, Debug Tool

IMS | DB2
CICS
WAS | MQ

z/OS

x86 PC running Linux

RDz user

RDz user

RDz user

ISPF user

RDz & ISPF user

RDz user
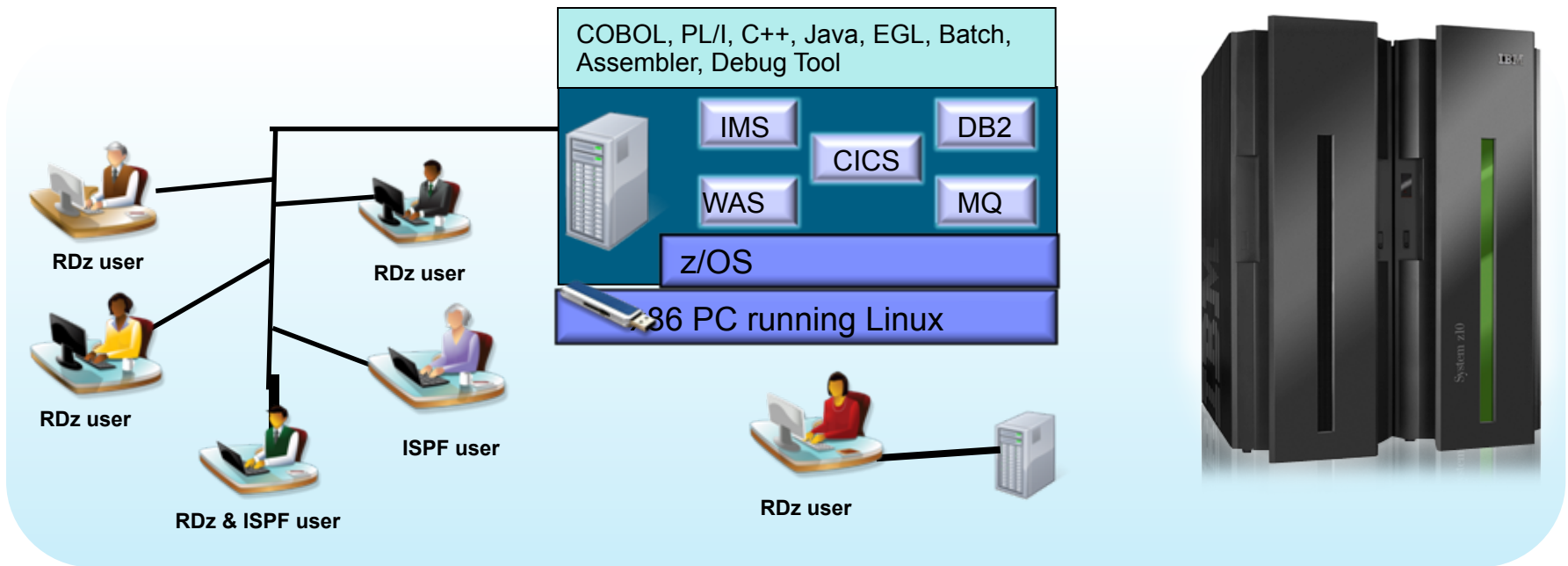
- ▪ Increase availability of z/OS testing environment and resources
  - ▪ Liberate developers to rapidly prototype new applications
  - ▪ Develop and test System z applications anywhere, anytime!
  - ▪ Eliminate costly delays by reducing dependencies on operations staff
- ▪ Improve quality and lower risk via automation, measurement, and collaboration
- ▪ Focus on what is required for the change at hand, then scale

SHARE
in Orlando 2015

Any process: Executable and repeatable
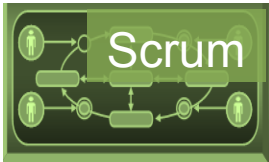
Use ONE tool to support both agile and non-agile
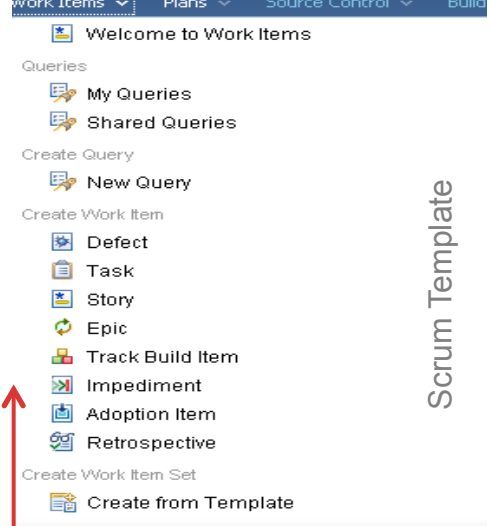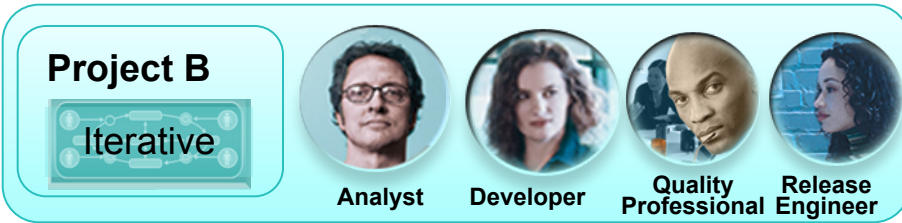
Process Architect
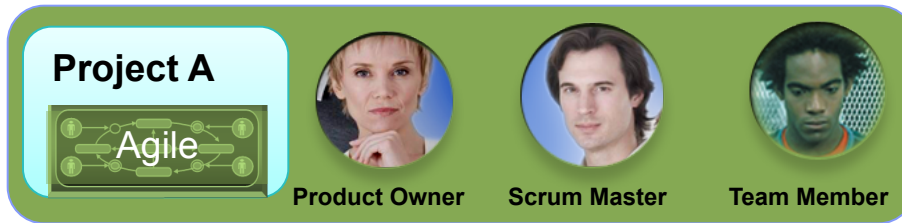
Project Manager

Variant #1
Scrum

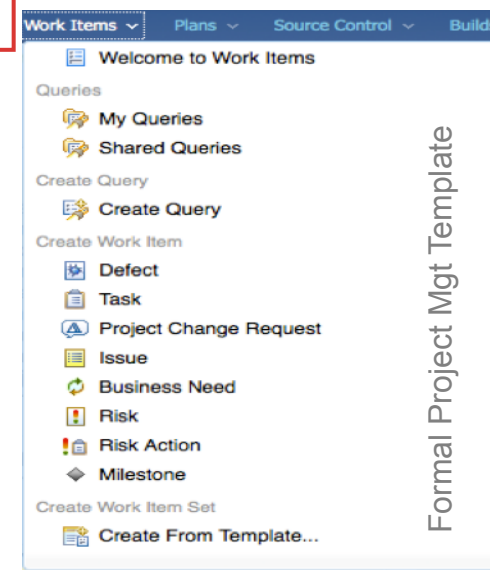*Variant #2*
Iterative

Variant #3
Waterfall

Project A
Agile
- Product Owner
- Scrum Master
- Team Member

Project B
Iterative
- Analyst
- Developer
- Quality Professional
- Release Engineer

Project C
Waterfall
- Analyst
- Developer
- Quality Professional
- Release Engineer

Work Items ▼   Plans ▼   Source Control ▼   Build

Welcome to Work Items

Queries
- My Queries
- Shared Queries

Create Query
- New Query

Create Work Item
- Defect
- Task
- Story
- Epic
- Track Build Item
- Impediment
- Adoption Item
- Retrospective

Create Work Item Set
- Create from Template

Scrum Template

Work Items ▼   Plans ▼   Source Control ▼   Builds

Welcome to Work Items

Queries
- My Queries
- Shared Queries

Create Query
- Create Query

Create Work Item
- Defect
- Task
- Project Change Request
- Issue
- Business Need
- Risk
- Risk Action
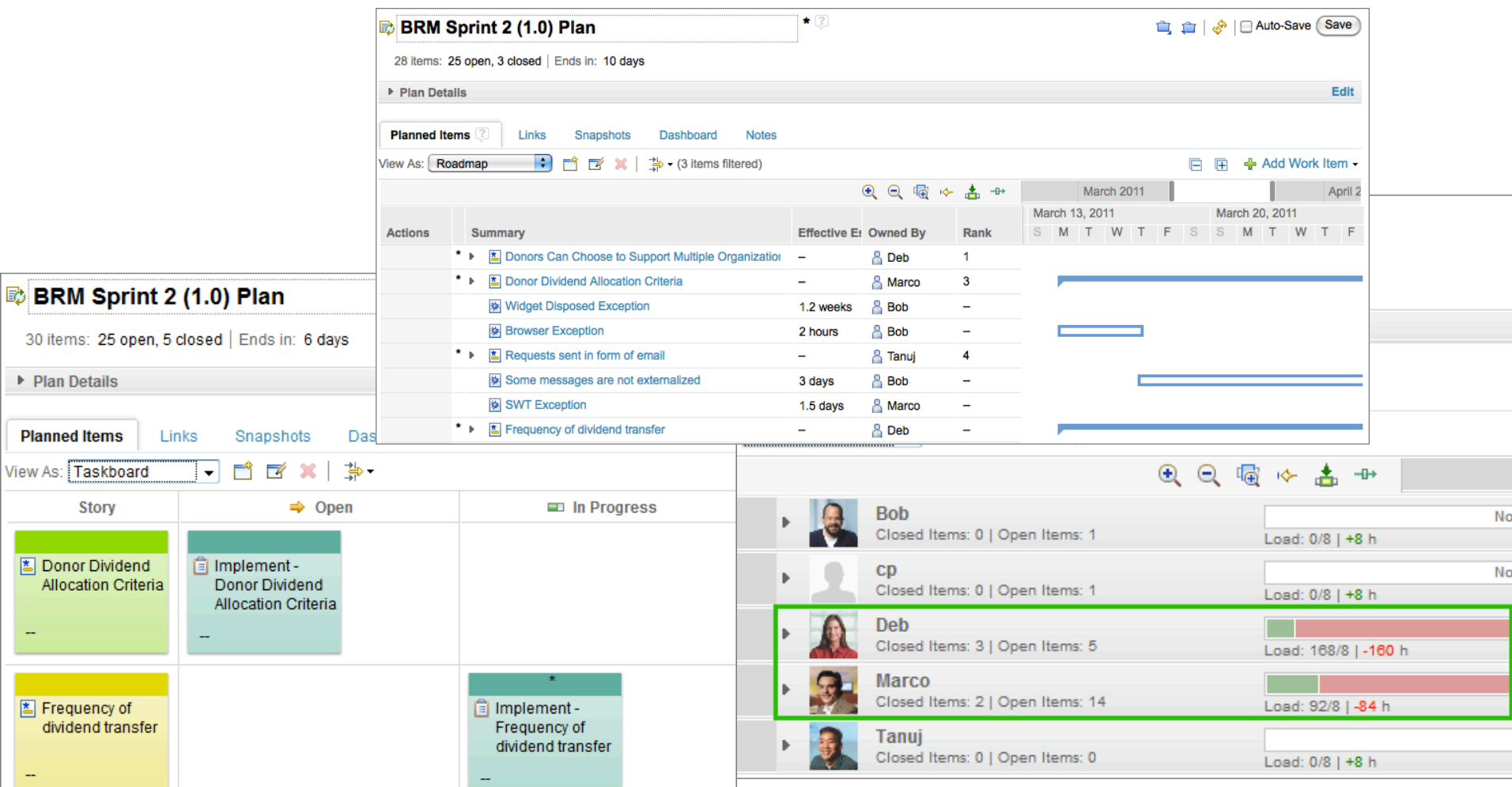- Milestone

Create Work Item Set
- Create From Template...

Formal Project Mgt Template

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

SHARE in Orlando 2015

Multiple plan views facilitate continuous planning

Progress Tracking: live status for everyone

Story Status Current Iteration
Shows the status of all stories planned for the current iteration

Foundation - Story Status (61) Status
- Testing
- Implementing
- Invalid
- Done
- Ready to Test
- New

(0 5 10 15 20 25 30 35 40 45 50)
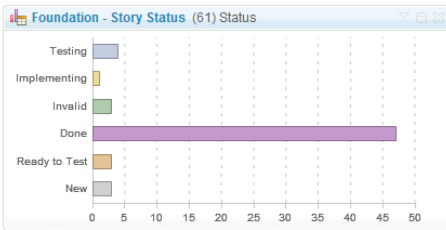
Foundation - Current Stories (61)
- 155918: Provide TVT testcases and testcase updates to TVT team
- 155399: [Repository - M14] Performance, quality and robustness enhancements
- 154407: [Continuation] Manage and repair DB inconsistencies during 3.0.1 M14
- 155024: Track the fixing of hangs occurring on shutdown of the CLM server
- 155033: Explore servlet dispatching enablement as a backup option
- 155030: [Application SDK - 14] Performance, quality and robustness enhancements
- 155120: LPA Web UI fixes and polish
- 111848: User can validate consistency of configuration information across contexts
- 155168: Fix bugs and improve the quality of LPA services
- 155624: update JREs for 3.0 iFix1 and 3.0.1

Page 1 of 7

CCM - Story Status (11) Status
- Testing
- Implementing
- New
- Ready to Implement

(0 1 2 3 4 5 6 7)

CCM - Current Stories (11)
- 150244: address "run as administrator" issues on win7
- 142127: As a user I want to see all extended state groups in taskboards
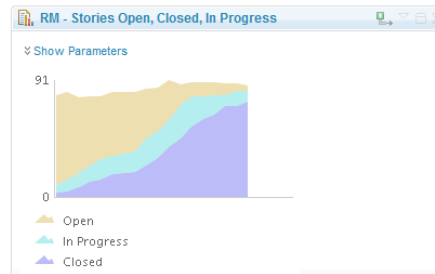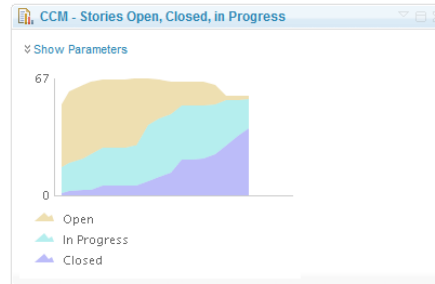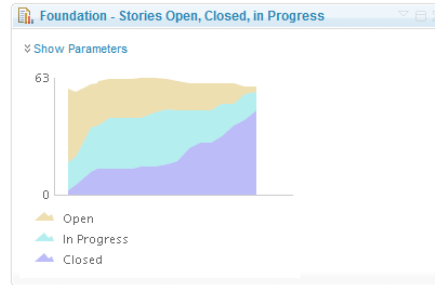- 150856: Improve the performance of loading a work item

Burndown Current Iteration
Shows the remaining amount of estimated work in hours of work items planned for the current iteration.

Foundation - Burndown
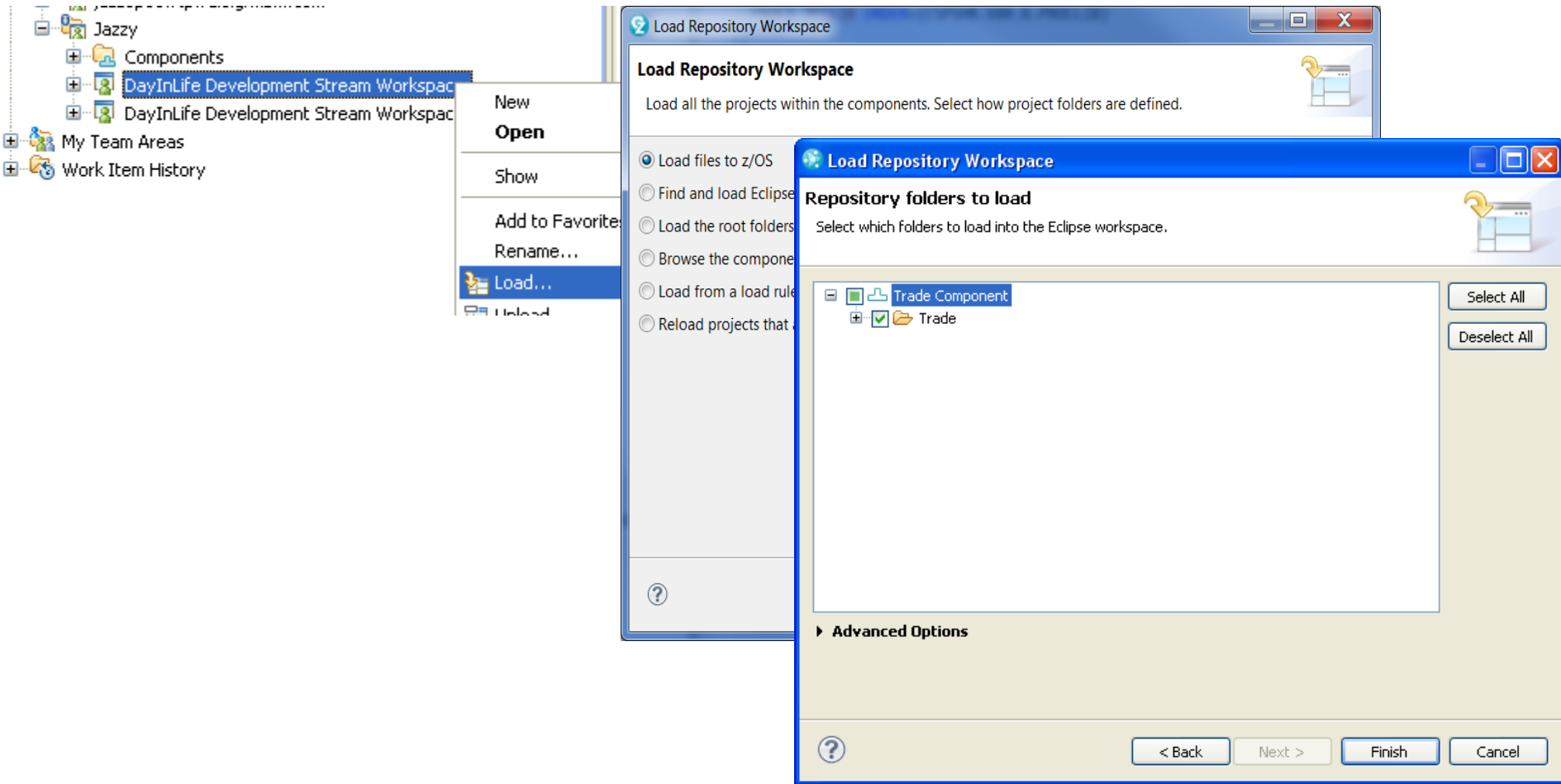⌄ Show Parameters

1,942

0

- Remaining Work
- Planned Work
- Ideal

CCM - Burndown
⌄ Show Parameters

2,808

0

- Remaining Work
- Planned Work
- Ideal

Burndown RM
⌄ Show Parameters

3,932
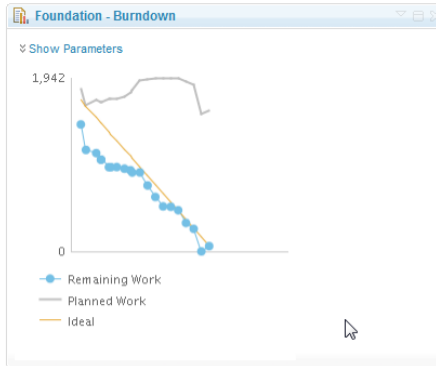
Stories Open/Closed/In Progress
Shows the number of stories which are open, in progress, done during the iteration.

Foundation - Stories Open, Closed, in Progress
⌄ Show Parameters

63

0

- Open
- In Progress
- Closed

CCM - Stories Open, Closed, in Progress
⌄ Show Parameters

67

0

- Open
- In Progress
- Closed

RM - Stories Open, Closed, In Progress
⌄ Show Parameters

91

0

- Open
- In Progress
- Closed

Load the source artifacts

RDz and RTC together

RDz Projects View shows Loaded Projects

RDz understands z/OS Development
- Rich, Language Specific Editors
- Enterprise web services
- Service Flow Modeler
- Code review
- Unit testing
- program analysis/control flow
- …
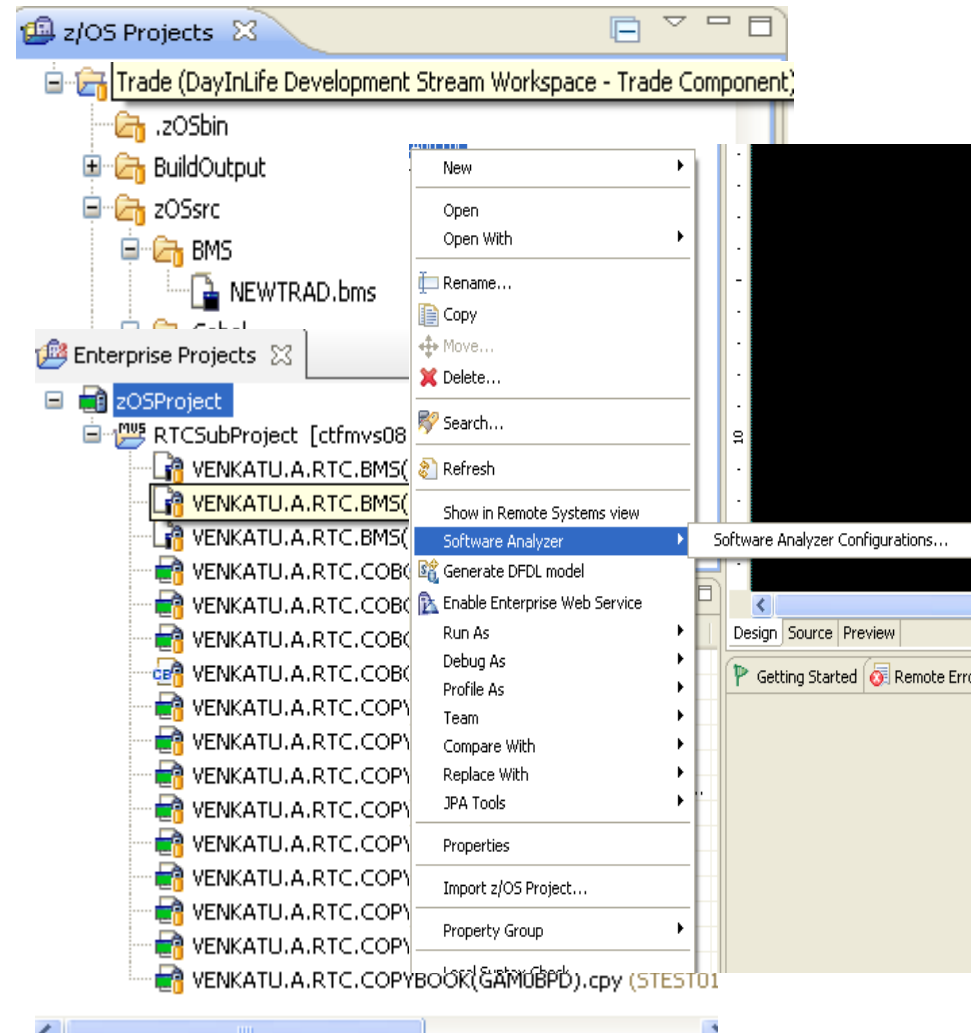
Navigate datasets and jobs on zOS

- Connect to multiple hosts concurrently

- Respects existing security configurations and user IDs

- Search, filter, browse, edit, compare, migrate, and allocate new MVS datasets and USS files

- Copy source code, members, or datasets between systems with a few mouse clicks.
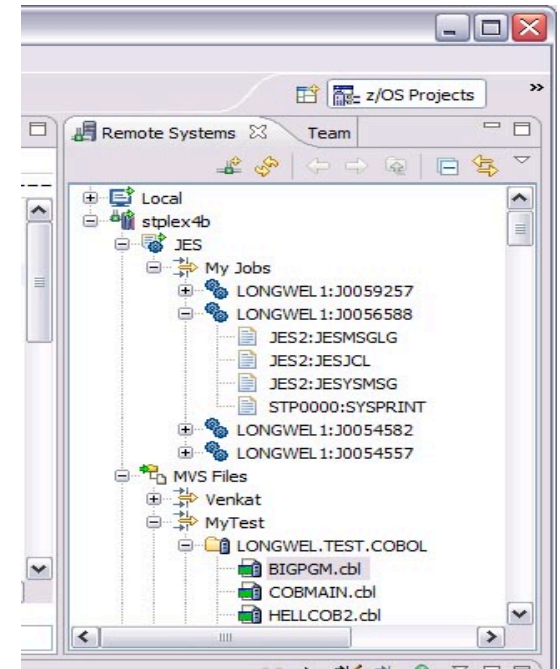
- Access JES queues submit jobs, view job state, and open output spools

- Submit TSO or USS commands

- Add datasets and members into projects to group applications and work items together logically

- Open an emulator in the IDE to configured hosts
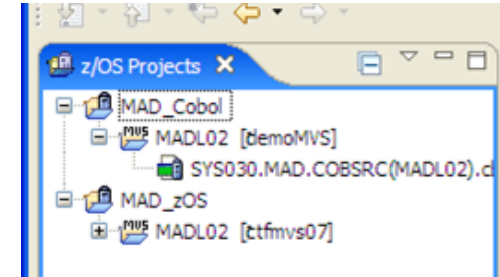
Edit capabilities in RDz

RDz at a high level has 2 types of editors

- COBOL, PLI, and JCL advanced editors
  - Based on the Eclipse editor infrastructure
  - Quick fixes, hyper-linking, hover
  - Easy navigation between edit sessions
  - Real time syntax checking, content assist, key word highlighting

- LPEX Editor
  - COBOL, PLI, HLASM, JCL, C/C++, REXX
  - ISPF-like experience including prefix commands, command line, look and feel
  - Supports advanced edit functions for COBOL, PLI and HLASM like real time syntax checking, content assist

COBOL, PL/I and JCL Editor improvements

Quick Outline

- Press Ctrl+O to activate

Editor Productivity Features

Real time syntax checking without code compile or save

JCL Editor Content Assist

Keyword syntax proposals
- Press Ctrl+SPACE to activate

Enhanced Application Quality & Structure Analysis

Application Analysis

- Control flow diagrams for COBOL and PL/1 programs,
    - *Graphical representation of the program flow with links to the source*
- Helps identify and highlight potential unreachable code
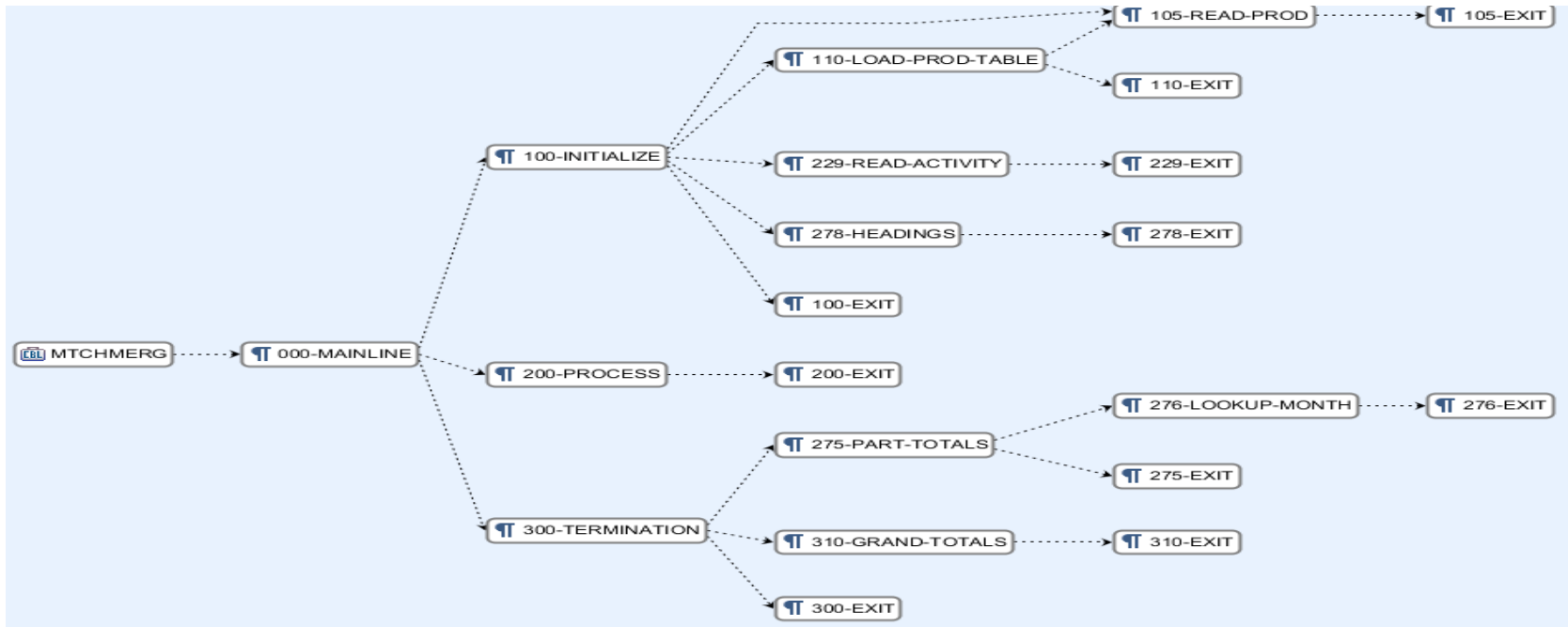
Enhanced Structure Analysis – Data Element Table

- A table representation of the user-defined data items and symbols in a program
  - Hyperlinks in the table are integrated with the editor allowing easy access to the declaration of the data items.

- Generated by showing the "symbol table" generated when RDz real-time syntax check parses the program

Data Element Table ✕

Showing data elements from WARDRPT.cbl   Search: [   ]

Data Item Name:	Data Type:	Level:	Top-Level Item:	Declaration:	Initial Value:	Line ...	Reference count:	Full Declaration:
PATLISTEST-S-ID	Data	10	PATIENT-MASTER-REC	PIC X(08)		378	0	10 PATLISTEST-S-ID PIC X
PATMSTR	File Descriptor	0	PATMSTR			116	4	FD PATMSTR RECORD CO.
PATMSTR-FOUND	Data	88	FILE-STATUS-CODES			134	1	88 PATMSTR-FOUND VALUE "0
PATMSTR-KEY	Data	5	PATMSTR	PIC X(06)		120	2	05 PATMSTR-KEY PIC X(06).
PATMSTR-REC	Data	1	PATMSTR			119	1	01 PATMSTR-REC.
PATMSTR-STATUS	Data	5	FILE-STATUS-CODES	PIC X(2)		133	3	05 PATMSTR-STATUS PIC X
PATPERSN	File Descriptor	0	PATPERSN			123	4	FD PATPERSN RECORD CO
PATPERSN-FOUND	Data	88	FILE-STATUS-CODES			136	1	88 PATPERSN-FOUND VALUE ".
PATPERSN-KEY	Data	5	PATPERSN	PIC X(06)		127	2	05 PATPERSN-KEY PIC X(06).
PATPERSN-REC	Data	1	PATPERSN			126	2	01 PATPERSN-REC.
PATPERSN-STATUS	Data	5	FILE-STATUS-CODES	PIC X(2)		135	3	05 PATPERSN-STATUS PIC
PATSRCH	File Descriptor	0	PATSRCH			98	4	FD PATSRCH RECORDING
PAYMENT-METHOD-TYPE	Data	5	PATIENT-PERSONAL-...	PIC X(02)		313	0	05 PAYMENT-METHOD-TYPE
PEDIATRICS	Data	88	INPATIENT-DAILY-REC			157	0	88 PEDIATRICS VALUE "1010".

User Build from RDz

User build is supported in zComponent projects and remote z/OS projects

- Builds the file selected, supports Error feedback
- Uses RTC Language definitions and Translators to generate the right JCL

August 2015
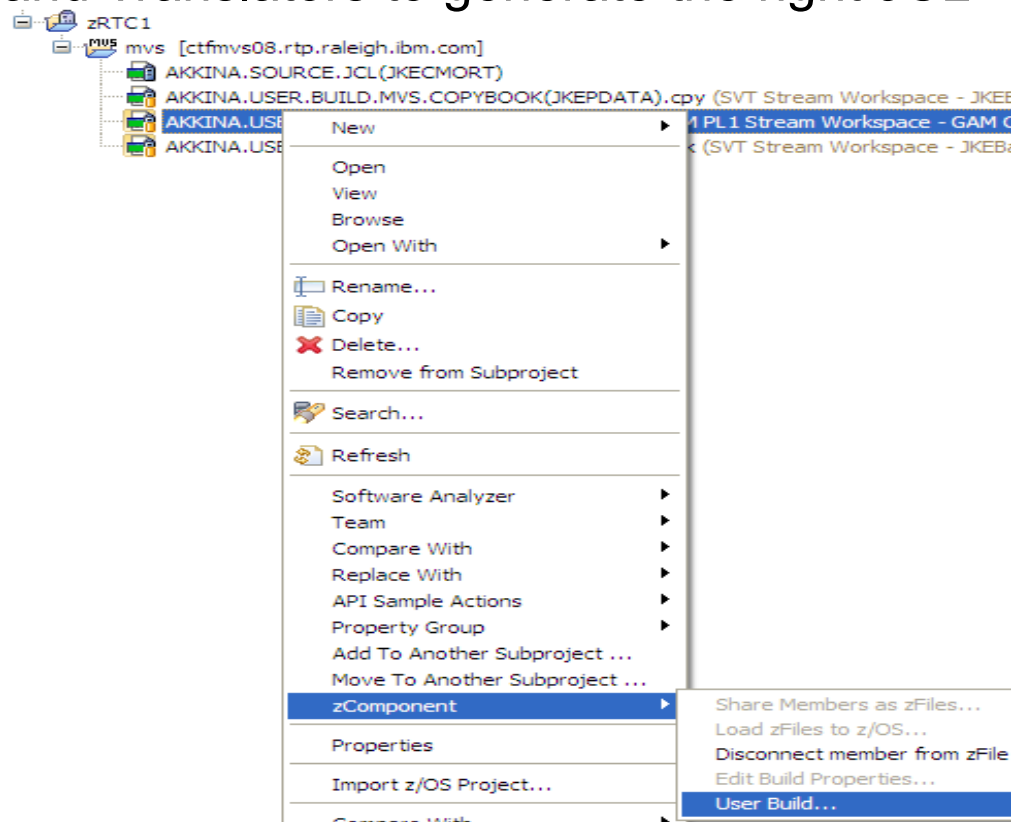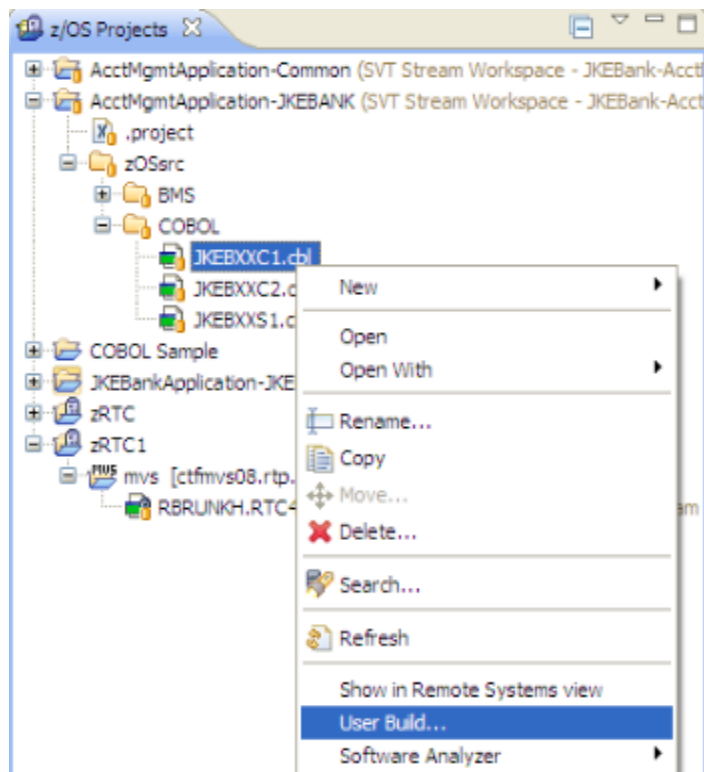
Pending Changes

Need fine-grained SCM control?
- Check out the pending changes view

Eclipse Rich Client:
- Check in, deliver, accept changes
- Suspend, resume, discard changes
- Replace, reload out-of-sync
- Resolve conflicts

Web Client:
- Open change sets and work
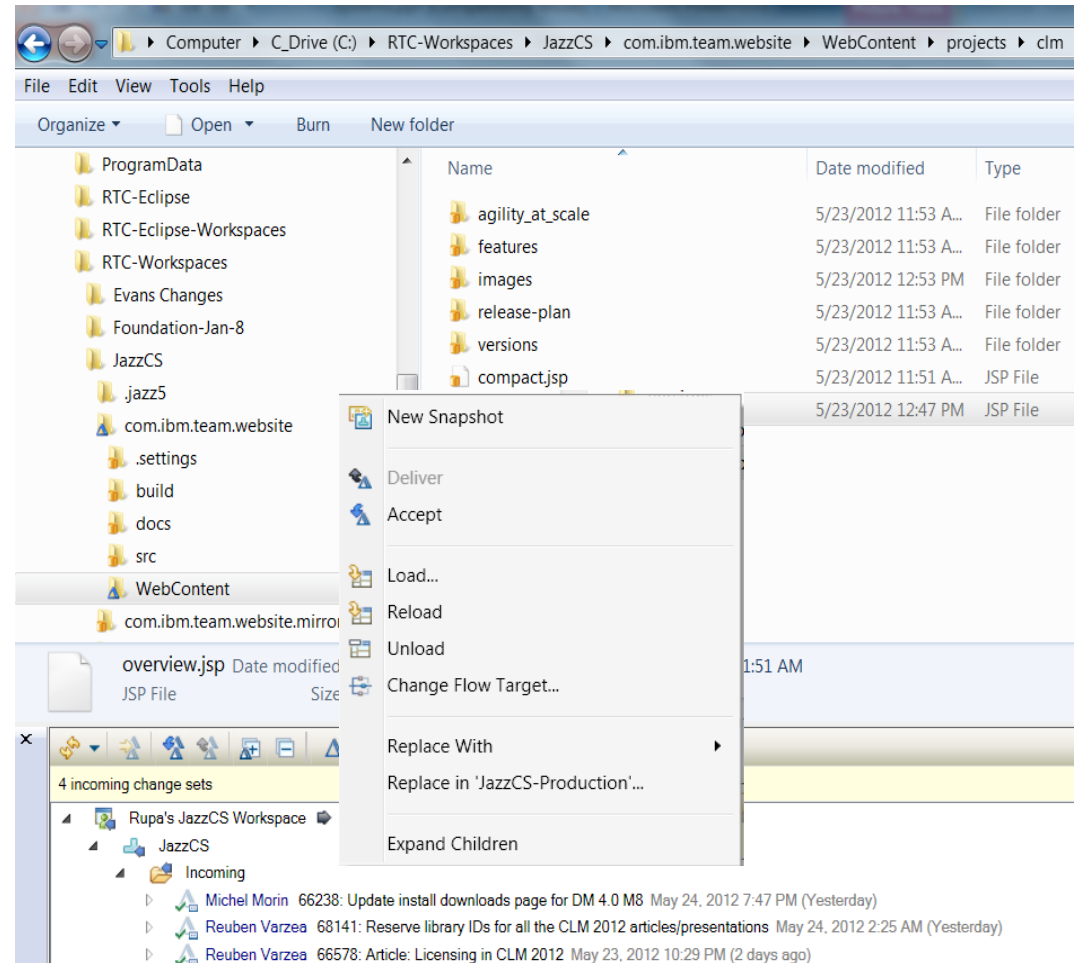
RDz Integrated Debugger

A GUI-based multi-platform, multi-language debugger

- ✓ Full asynchronous mode
  - ✓ Thread-level control of multi-threaded applications
- ✓ Automonitor support

RDz v9.0.1 Base Support**:**

- ✓ COBOL V5.1, V4, V3.4
- ✓ Batch, Batch IMS, Batch DB2, CICS 5.1, 4.2, 4.1
- ✓ Interactive Code coverage – Out of the box

RDz 9.1 new support:

- ✓ PLI v4.x, v3.9
- ✓ C/C++ V1R13, V2R1
- ✓ IMS TM
- ✓ DB2 Stored procedures
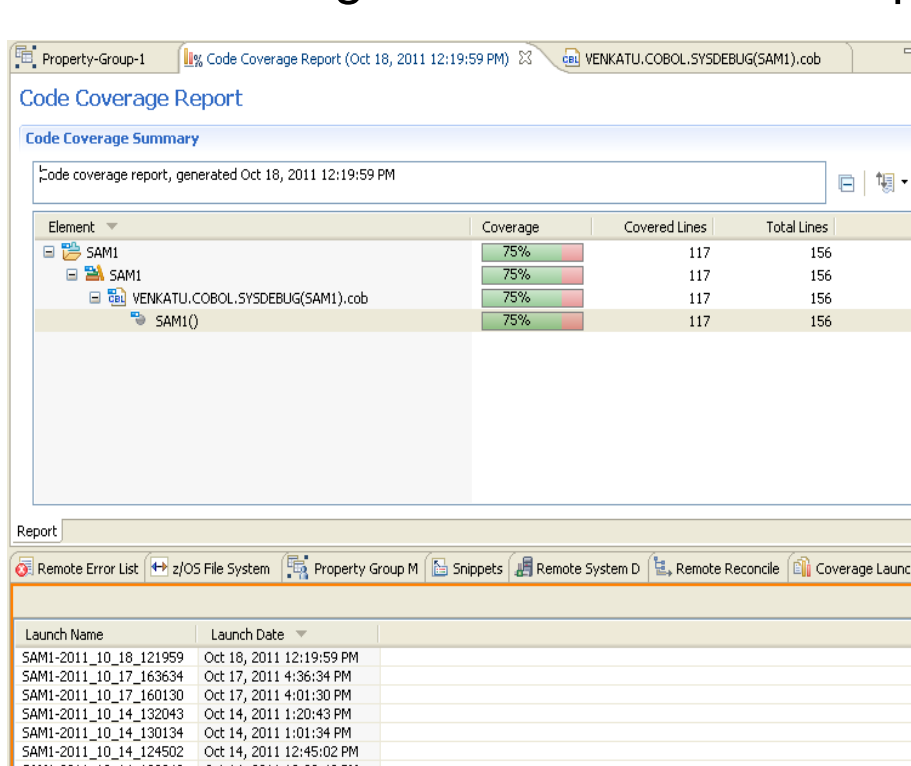
Debug Multiple Runtimes

Use the cross-platform debugger to debug end-to-end systems as they execute in the runtime

- CICS
- Batch
- Java

From the workstation:

- View executing source code
- Step through host code line-by-line
- Set breakpoints
- Alter working storage values
- Alter register values
- Etc…

Debug z/OS and distributed code

- in the same interface even stepping between runtimes and platforms!

Enhanced Application Quality – Code Coverage

Measure and report test coverage of an application
- Supports multiple languages including COBOL and Java
- Leverages the Integrated Debugger technology
- Indicates what source code was tested and remains to be tested
- View Coverage at module / source / procedure level

Predefined rules and templates

- for COBOL and PL/I applications
- Ensure corporate standard adherence
- Custom rules for COBOL and PL/I

Add Custom Rule

Select Template

Select a template to use for this rule

- An inline PERFORM statement cannot exceed the specified line number limit
- Avoid calling the specified routine
- Avoid including the specified copy book
- Avoid nesting IF statements deeper than the specified number of levels
- Avoid static calls except for the specified program name
- Avoid using the selected compiler directives
- EXEC SQL: Use a WHERE clause in selected statements
- Follow the specified naming convention for COBOL file names
- Procedure division statements cannot exceed the specified line number limit
- Use a scope terminator phrase with the specified COBOL statement type
- Use comments to describe all paragraphs
- Use comments to describe all sections
- Use the specified prefix with condition names

`< Back` `Next >` `Finish` `Cancel`

- COBOL Code Review [0/43]
  - Naming Conventions [0/1]
    - Use a program name that matches the source file name
  - Performance [0/9]
    - Avoid INITIALIZE statements. Use elementary MOVE statements or VALUE clauses.
    - Avoid OCCURS DEPENDING ON phrases
    - Avoid using subscripts to access a table. Use indexes.
    - **EXEC SQL: Avoid SELECT ***
    - **EXEC SQL: Use an ORDER BY clause when declaring a cursor**
    - **Specify 0 RECORDS for BLOCK CONTAINS clauses in file description entries**
    - Use an EVALUATE statement rather than a nested IF statement
    - **Use an odd number of digits in a COMP-3 or PACKED-DECIMAL data definition**
    - Use binary subscripts
  - Program Structures [0/33]
    - Avoid ACCEPT statements
    - Avoid ACCEPT statements containing FROM CONSOLE or FROM SYSIN
    - Avoid ALTER statements
    - Avoid CALL statements with a literal program name
    - **Avoid CANCEL statements**
    - Avoid COPY SUPPRESS statements
    - Avoid CORRESPONDING phrases
    - **Avoid DISPLAY statements containing UPON CONSOLE**
    - Avoid ENTRY statements
    - Avoid EXIT PROGRAM statements
    - Avoid GO TO statements
    - Avoid GO TO statements, except those that reference an EXIT paragraph
    - **Avoid IF without ELSE**
    - Avoid NEXT SENTENCE phrases
    - **Avoid PERFORM, except PERFORM section**
    - Avoid RESERVE clauses in FILE-CONTROL paragraphs
    - Avoid STOP RUN and STOP literal statements
    - Avoid THRU phrases in PERFORM statements
    - **Avoid using level-88 entries in data descriptions**
    - **Avoid using more than one EXIT statement per section**
    - **Avoid using SECTION in the procedure division**
    - **Avoid XML PARSE statements**
    - **EXEC CICS: Check EIBRESP after NOHANDLE**
    - EXEC CICS: Use DFHRESP to check the return value
    - EXEC CICS: Use the RESP option
    - **EXEC SQL: Check the value of SQLCODE after an EXEC SQL statement**
    - **Use an EXIT paragraph in each section**
    - Use a WHEN OTHER phrase with an EVALUATE statement

zUnit: Unit testing framework for z/OS

A Framework for developers to write:

- repeatable, self-checking unit tests, based on *xUnit*.

- xUnit a light-weight architecture to implement unit test frameworks.

  - JUnit, for example, is a very popular instance of the xUnit architecture.

- **zUnit** is an xUnit instance for System z

- zUnit encourages **continuous integration and continuous testing**

  - for System z Application development and maintenance

USER.ZUNIT(TESTCASE)

```
┌─────────────────────────────────┐        zUnit Test Runner API
│  ┌─────────────────────────┐    │      ┌────────────────────────────┐
│  │ TESTCASE[1](...)        │    │      │  ┌──────────────────────┐  │
│  └─────────────────────────┘    │      │  │ ZXUTCINI(...)        │  │
│  ┌─────────────────────────┐    │      │  └──────────────────────┘  │
│  │ ADDTESTS[2](...)        │    │      │  ┌──────────────────────┐  │
│  └─────────────────────────┘    │      │  │ ZXUTCADD(...)        │  │
│  ┌─────────────────────────┐    │      │  └──────────────────────┘  │
│  │ SETUP[2](...)           │    │      │  ┌──────────────────────┐  │
│  └─────────────────────────┘    │      │  │ ZXUASTFM(...)        │  │
│  ┌─────────────────────────┐    │      │  └──────────────────────┘  │
│  │ TEARDOWN[2](...)        │    │      │  ┌──────────────────────┐  │
│  └─────────────────────────┘    │      │  │ ZXUASTFA(...)        │  │
│  ┌─────────────────────────┐    │      │  └──────────────────────┘  │
│  │ TEST001[2](...)         │    │      └────────────────────────────┘
│  └─────────────────────────┘    │
│        . . .                    │
│  ┌─────────────────────────┐    │
│  │ TESTnnn[2](...)         │    │
│  └─────────────────────────┘    │
└─────────────────────────────────┘
```
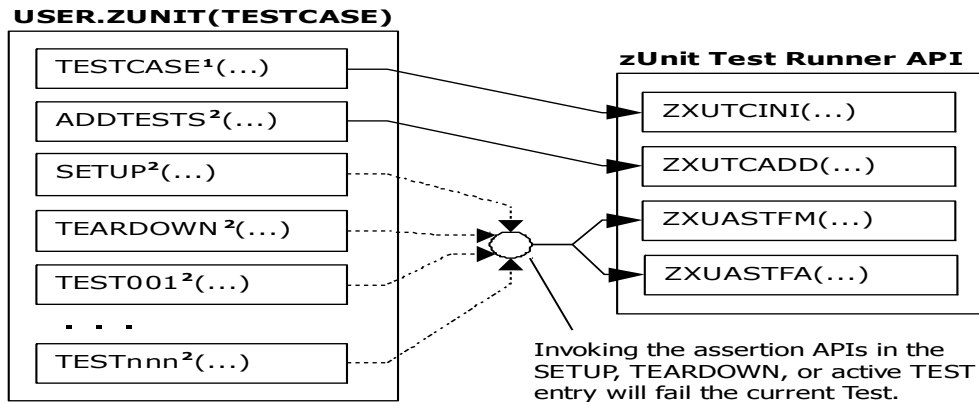
Invoking the assertion APIs in the SETUP, TEARDOWN, or active TEST entry will fail the current Test.

[1]Language-specific details:
- In COBOL, this is the first program appearing in the Test Case source file and it will be invoked by the Test Runner for Test Case initialization.
- In PL/I, the is the procedure declared with option(fetchable) in the Test Case source file and it will be invoked by the Test Runner for Test Case initialization.

[2]Language-specific details:
- In COBOL, these are expected to be subprograms (non-nested and therefore compatible with FUNCTION-POINTER).
- In PL/I, these are expected to be internal procedures that are declared at the package level (non-nested).

IBM Problem Determination Tools for z/OS (PD Tools)

Application Performance Analyzer for z/OS

- Identify performance and response time problems. Assists in reducing resource consumptions.

Debug Tool for z/OS

- Interactive debugging, understanding execution flow, COBOL modernization, code coverage and much more.

Fault Analyzer for z/OS

- Abend analysis at source-level reporting to speed up the problem analysis steps.

File Manager for z/OS

- Data manipulation on z/OS including data in DB2, IMS, CICS, MQ, data sets and HFS files.

IBM Problem Determination Tools Modernization Solution Pack
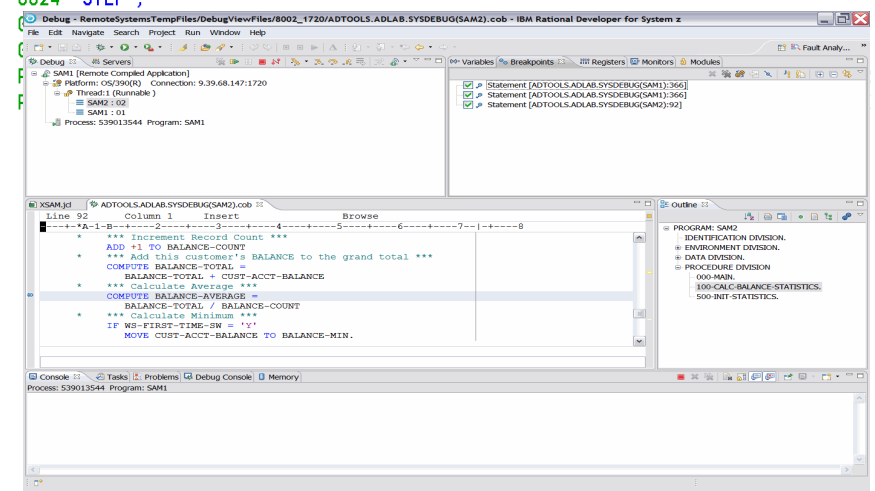
IBM Debug Tool Version 13

Debug Tool can help you increase debugging efficiencies and reduce application development cycle times.

▪ Program testing and analysis aid that helps you examine, monitor, and control the execution of applications on z/OS (CICS/DB2/IMS/COBOL/PLI/ASM,C/C++/ASM/JAVA w Toolkit)

- Proven 3270 and eclipse interfaces
- Improved Code Coverage facilities for COBOL and PL/1
- Automatic start of IMS MPP region with dynamic transaction routing
- Playback support in RD/z
- Support for new instruction set available with z13 hardware
- Support for Enterprise COBOL V5.2 and V5,1 compilers
- Support for Enterprise PL/I V4.5 compiler
- Support for z/OS XL C/C++ V2.1 compilers
- Support for CICS Transaction Server V5.2
- Supporting CICS TS V5.3 Open Beta program
- DB2 V11 for z/OS
- IMS 13 Transaction and Database Servers
- WAS for z/OS V8.5
- MQ for z/OS V8.0

Debug Tool: Continuous advancements

Technical Preview

Web: Code coverage

- Browser interface for reviewing code coverage report.
- Quickly understand coverage of regression testing effort.
- Drill down to source line level for detailed analysis.
- Code coverage services can be accessed via APIs.

Eclipse: Load Module Analyzer

- Analyzes MVS load modules or program objects to determine the language translator (compiler or assembler) used to generate the object for each CSECT.
- Invoke Load Module Analyzer from Eclipse environment.

ISPF: JCL Wizard

- An ISPF edit macro that can be used to modify a JCL or procedure member, creating the statements to invoke Debug Tool in various environments.
- Creates statements to invoke Debug Tool for z/OS for:
    - Terminal Interface Manager, Mainframe Interface, Remote GUI.

Summary and chronological Information
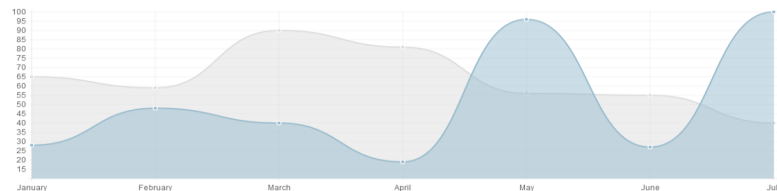
Connected to: TLBA07ME.TOROLAB.IBM.COM

Groups		Users	
GUI	92%	glen	92%
Server	87%	jesse	87%
Regression	55%	francisco	55%

Working with CC reports expanding on a user: Annotated and color coded


```
71
72  e              MOVE SPACE TO TBND003F-DATA-OUT.
73
74  e              IF TBND003F-DATA-IN IS NUMERIC
75  e                  IF TBND003F-DATA-IN u 1 OR TBND003F-DATA-IN e 26
76  u                      MOVE 'INVALID TABLE 2 POINTER' TO TBND003X-MSG
77  u                      MOVE 341 TO RETURN-CODE
78                      ELSE
79  e                      MOVE WK-LETTER (TBND003F-DATA-IN) TO
80                              TBND003F-DATA-OUT
81  e                          MOVE 0 TO RETURN-CODE
82                      END-IF
83                  ELSE
84  u                      MOVE 'TABLE 2 PTR NOT NUMERIC' TO TBND003X-MSG
```

```
Lookup   z/OS Job   DTSP Server Profile   DTCN Server Profiles   Instrument JCL for Debug Tool Debugging   Load Module Analyzer   LMA Report Viewer
LMAReport_ELIN.CC5.LOADPDSE   LMAReport_ELIN.CC5.LOADPDSE
DFS     Active (IMS)          IMS
DSN     Active (DB2)          DB2
EDC     Active (C/C++)        C/C++
EZA     Active (TCP/IP)       TCP/IP
IBM     Active (PL/I)         PL/I
IEDC    Active (C/C++)        C/C++
IEL     Active (PL/I)         PL/I
IEN     Active (PL/I)         PL/I
IGY     Active (COBOL)        COBOL
IGZ     Active (COBOL)        COBOL
IIBM    Active (PL/I)         PL/I
ILB     Active (OS/VS COBOL)  OS/VS COBOL

1 5655-Q10   Debug Tool Version 13 Release 1.2   Load Module Analyzer   2014/09/09  16:50  Page 3
EQAPGMNM:

0000000 (?)              (No IDR data available)
```

Enterprise COBOL V5.1 & V5.2 Exploitation
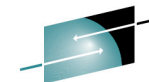
Enterprise COBOL V5.2

- Provides easy migration from COBOL V5.1
- Includes all enhancements delivered in COBOL V5.1 PTFs
  - Restored migration features, performance features, new programing features
- Provides Day 1 support for z13 processor
  - Instruction scheduler tuned to new micro architecture at ARCH 11
  - Expanded use of Decimal Floating Point for PACKED-DECIMAL data
  - Uses new SIMD instructions for INSPECT TALLYING or REPLACING statements
- Improves Application Performance
  - Well-structured, compute intensive batch applications running on z13 (compiled with Enterprise COBOL V5.2) have shown CPU time reduction of up to 14% over the same applications running on zEC12 (compiled with the GA release of Enterprise COBOL V5.1) [1]

[1] Results are based on an internal compute-intensive test suite. Performance results from other applications may vary.

developerWorks Article: **Prepare to upgrade to Enterprise COBOL for z/OS V5 Enterprise COBOL for z/OS V5.1.1 Migration Guide**

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

PD Tools Exploitation

- The new DWARF format side file of COBOL V5.1 brings a significant advantage for its users especially when performing problem diagnosis using tools like PD Tools:
  - Program information required for problem diagnosis is stored in the same file as the load module.
  - This simplifies the management of side files as it is no longer required to maintain load modules and matching side files separately.
  - Since DWARF information is created and stored while link edit the program, it eliminates the possibility of side file mismatch.
  - No performance penalty for selecting the DWARF option. DWARF information is not loaded into storage while program executes normally. It is only read from the file when it is required by problem diagnosis tools like PD Tools.
- Application Performance Analyzer for z/OS, Debug Tool for z/OS and Fault Analyzer for z/OS fully exploit the new DWARF format COBOL V5.1 side file format.
- All tools supports new z13 hardware instructions exploited by the compiler.
- File Manager for z/OS continues to support COBOL V5.1 level COPYBOOKs.

August 2015

Improved Productivity

Application is abending. What do I do?

Simple S0C8 in COBOL program analysis

Steps without FA:

1. Capture CEEDUMP or MVS dump
2. Determine abend offset into program from dump traceback
3. Obtain a matching compiler listing – need to make sure it really matches!
4. Using the listing, identify the matching source line for the abend offset
5. Identify data fields involved and their offsets into working storage
6. Validate the content of each data field in the dump to determine the one(s) in error
7. Fix the problem

Do I really want to do this?

Improved Productivity

Solution provided by Fault Analyzer report

```
pthfmd2:2820/KENICHI.HISTORY(F00002)-Report ⊠
 1⊖
 2  Module COBTSE, program COBTSE, source line # 17: Abend S0CB (Decimal-Divide Exception)
 3⊖
 4   I B M   F A U L T   A N A L Y Z E R   S Y N O P S I S
 5
 6
 7  A system abend 0CB occurred in module COBTSE program COBTSE at offset X'458'.
 8
 9  A program-interruption code 000B (Decimal-Divide Exception) is associated with
10  this abend and indicates that:
11
12    The divisor was zero in a signed decimal division.
13
14  The cause of the failure was program COBTSE in module COBTSE.  The COBOL source
15  code that immediately preceded the failure was:
16
17    Source
18    Line #
19    ------
20    000017             DIVIDE NUM1 BY NUM2 GIVING NUM3.
21
22  The COBOL source code for data fields involved in the failure:
23
24    Source
25    Line #
26    ------
27    000006      01 NUM1 PIC 99 VALUE ZERO.
28    000007      01 NUM2 PIC 99 VALUE ZERO.
29    000008      01 NUM3 PIC 999 VALUE ZERO.
30
31  Data field values at time of abend:
32
33    NUM1 = 10
34    NUM2 = 0  *** Cause of error ***
35    NUM3 = 23
```

Gives you an explanation of failure and pinpoint erroneous source line #

Displays source information involving in the error

Showing you precisely what data needing the fix

Fault Analyzer: Continuous advancements

COBOL Explorer: Post-mortem debugger

NEW

- **Want to understand the execution path which led to a failure from a dump?**
- **Want to see how a variable changed and contributed to a failure?**
- COBOL Explorer is a new feature of Fault Analyzer which lets you interactively navigate through your application from the point of failure, examining variable ref/def

Start

Main program

```
. . . .

MOVE 0 AVALUE.
MOVE 100 ATOTAL.

CALL CALC USING AVALUE ATOTAL.

. . .
```

Statement you need to fix!!!

Subroutine CALC

```
. . . .

LINKAGE SECTION.
01 X PIC 9(4).
01 Y PIC 9(4).


. . . .
DIVIDE Y BY X GIVING Z.

. . .
```

Point of Failure
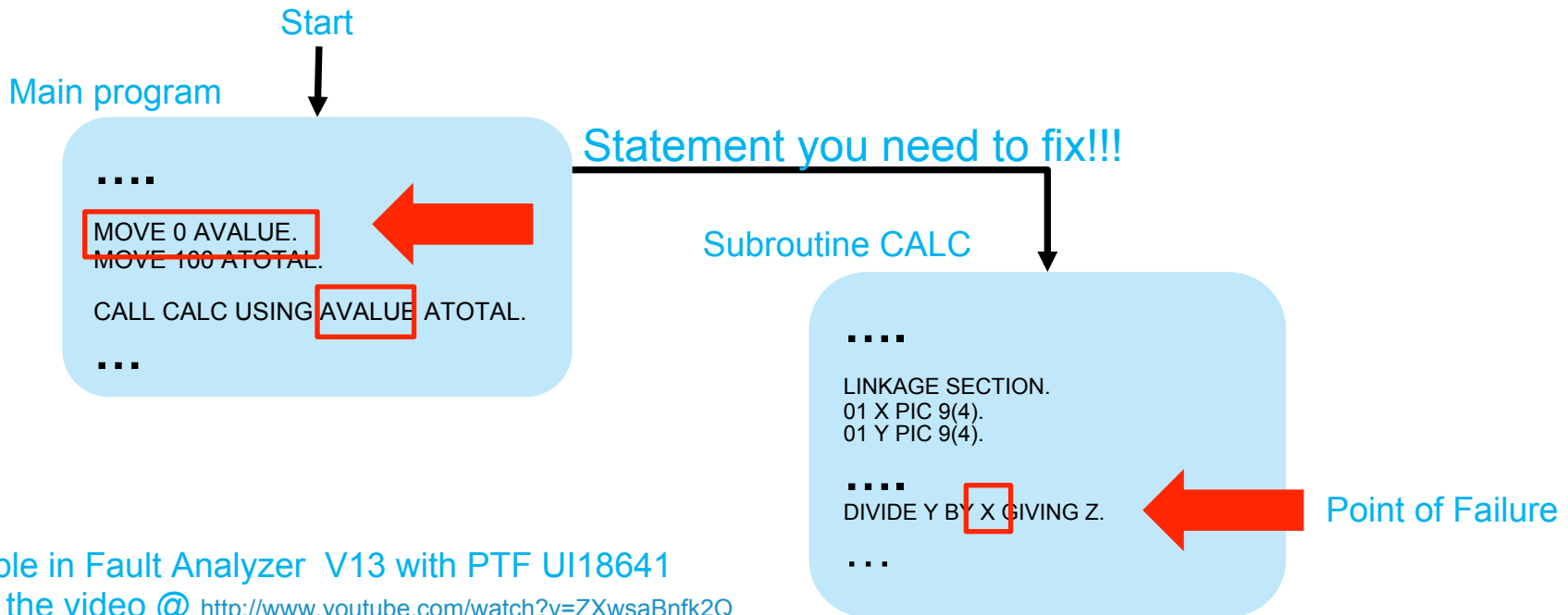
Available in Fault Analyzer V13 with PTF UI18641
Watch the video @ http://www.youtube.com/watch?v=ZXwsaBnfk2Q

SHARE in Orlando 2015

File Manager: Continuous advancements
Data scrambling – protecting sensitive information

File Manager in conjunction with Optim ODPP APIs provides powerful data scrambling for your enterprise data residing in a variety of locations on z/OS.

- Work with complex data structures such as SMF records and CSC Hogan Systems data.
- Support for most data sources found on z/OS: DB2, IMS, Data Sets, HFS files, WebSphere MQ, and CICS TS.
- Take advantage of powerful File Manager's copy function.

APA: Continuous advancements
How to fix performance related problems?

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

August 2015

APA: Continuous advancements
How to fix performance related problems?

Summary

Many companies spend >70% on keeping lights on, and that amount is increasing

IT organizations have problems modifying applications at speed of business

IBM provides a structured approach to incrementally modernize your portfolio

- based on business priorities

Change without a plan is chaos

A Plan without change is stagnation

Business goals change

- applications need to change to address goals

Continual renewal is required

- tools help to guide, govern, drive, and accomplish change

Getting started

Next steps to modernize your enterprise applications

www.ibm.com/rational/modernization

- **Try latest System z software for free**
- **Sign up for free web-based training**

- **Join IBM Rational Cafe Communities**

- **Get prescriptive service solutions**

- **Success stories**

- **Latest news on System z twitter**

- **Latest customer videos**

- **Latest skills: System z job board**

Empower

Unify

Revitalize

Enterprise Modernization

Optimize

Enabling Product and Service Innovation

jazz

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

August 2015

धन्यवाद
Hindi

多謝
Traditional Chinese

Grazie
Italian

ขอบคุณ
Thai

Gracias
Spanish

Merci
French

Спасибо
Russian

شكراً
Arabic

Obrigado
Brazilian Portuguese

Danke
German

多谢
Simplified Chinese

நன்றி
Tamil

ありがとうございました
Japanese

감사합니다
Korean

Notices and Disclaimers

Notices and Disclaimers (con't)

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. IBM EXPRESSLY DISCLAIMS ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

- IBM, the IBM logo, ibm.com, Bluemix, Blueworks Live, CICS, Clearcase, DOORS®, Enterprise Document Management System™, Global Business Services ®, Global Technology Services ®, Information on Demand, ILOG, Maximo®, MQIntegrator®, MQSeries®, Netcool®, OMEGAMON, OpenPower, PureAnalytics™, PureApplication®, pureCluster™, PureCoverage®, PureData®, PureExperience®, PureFlex®, pureQuery®, pureScale®, PureSystems®, QRadar®, Rational®, Rhapsody®, SoDA, SPSS, StoredIQ, Tivoli®, Trusteer®, urban{code}®, Watson, WebSphere®, Worklight®, X-Force® and System z® z/OS, are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.