



IBM Software Group

Lowering the Total Cost of Ownership for Corporations doing Traditional z/OS Development, Maintenance and Support with Rational Developer for System z

Jon Sayles

RDz Technical Enablement

jsayles@us.ibm.com



Rational. software

Agenda

- IBM/Rational "Selective Offloading – Comprehensive Coverage"
- Additional Cost Savings
- Factors in Mainframe Chargeback Savings
- The Return-on-Investment Spreadsheet
- Working session

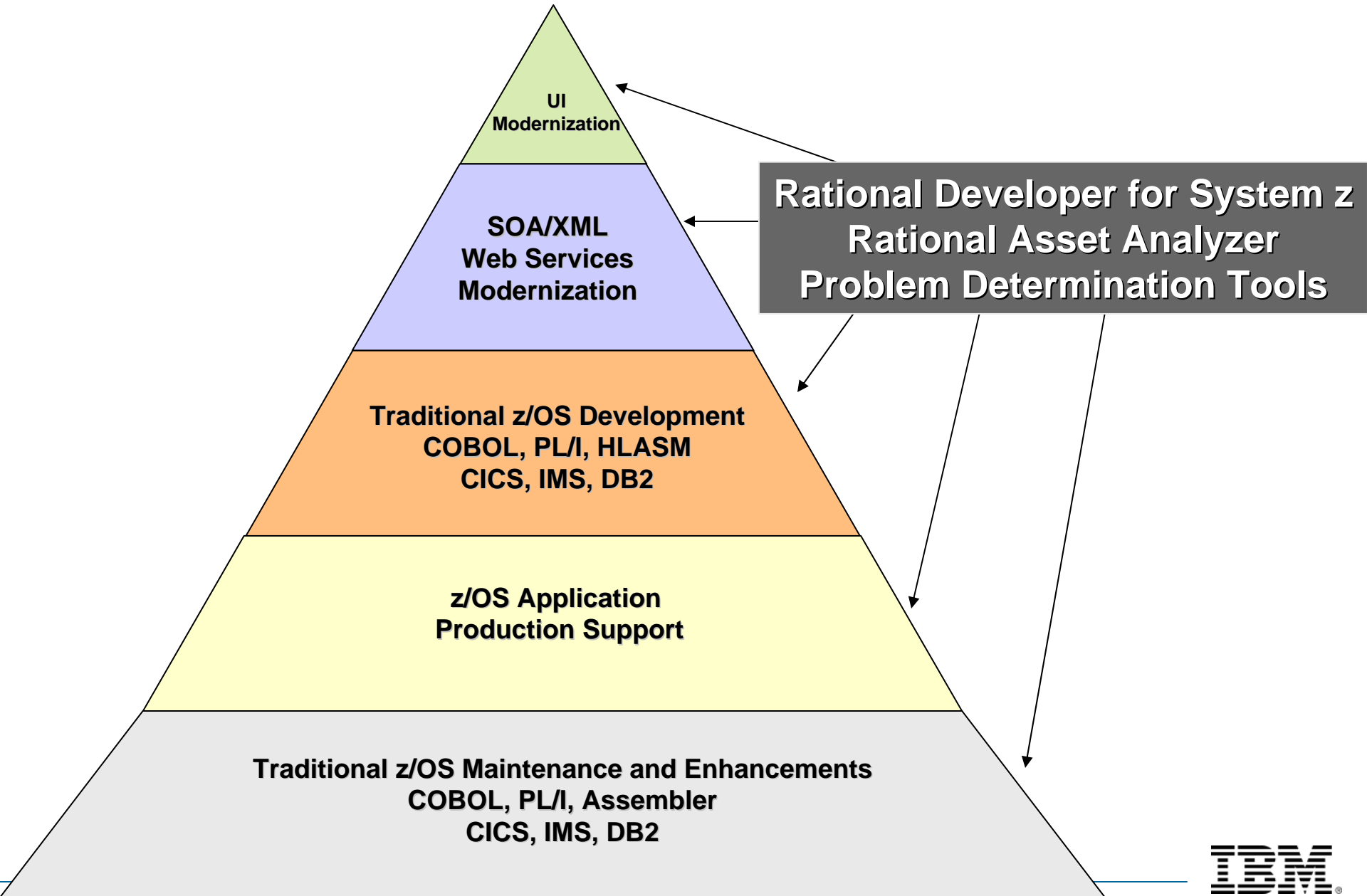
All performance data contained in this publication was obtained in the specific operating environment and under the conditions described in this publication and is presented as an illustration only.

Performance obtained in other operating environments may vary and customers should conduct their own testing.

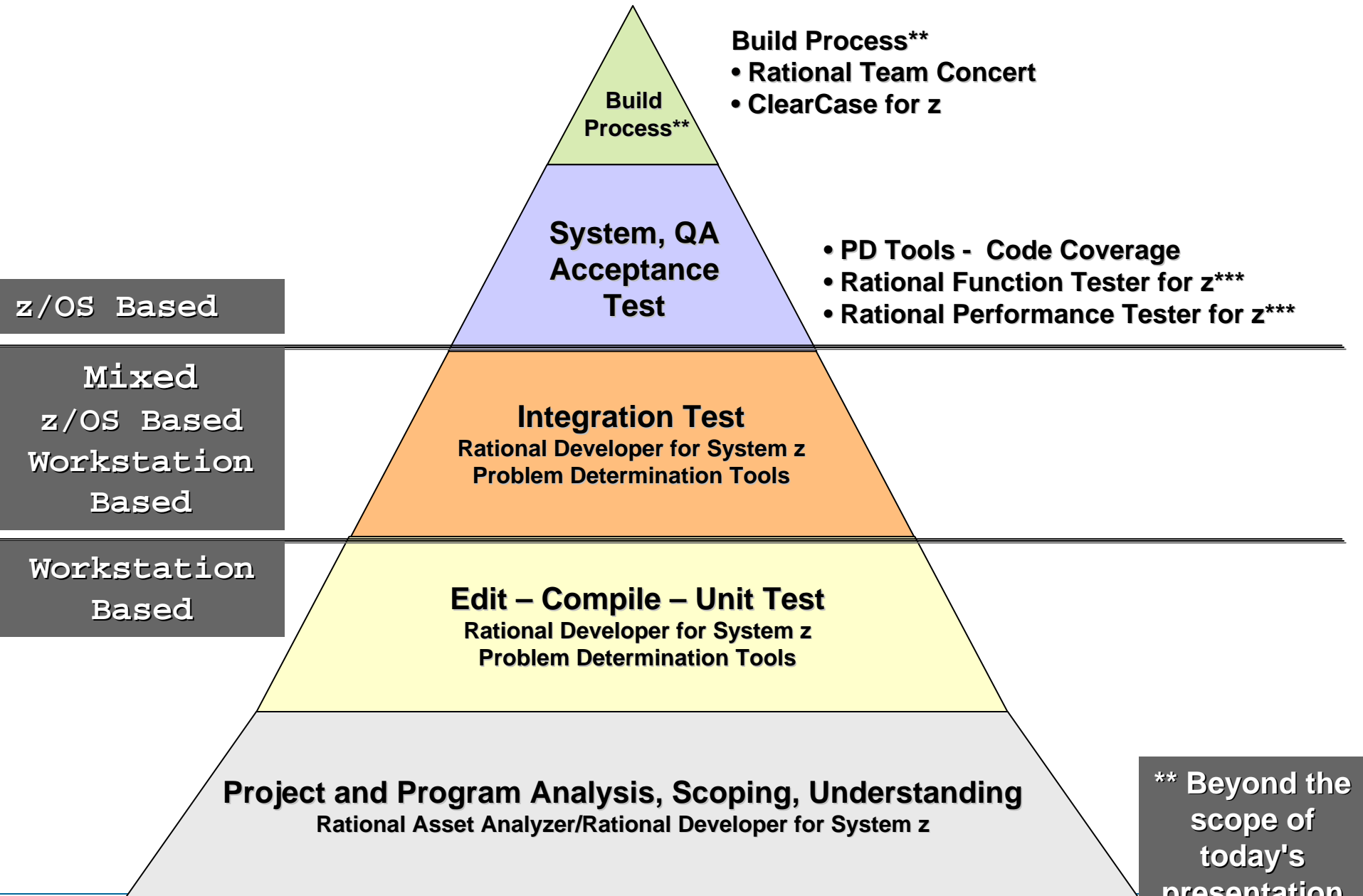
IBM/Rational's Selective Offloading/Comprehensive Coverage

- With RAA/RDz/PD Tools, you get the benefit of MIPS reduced development for:
 - ▶ COBOL, PL/I, C/C++, Assembler
 - ▶ JCL
 - ▶ BMS/MFS
 - ▶ SQL, DL/I
- **No offloading an entire z/OS run-time platform**
 - ▶ No Data and file logical extracts
 - ▶ No downloading and configuring Resource Definitions
 - ▶ No worries about called utilities or Assembler
 - ▶ In fact, you do not have to offload any artifacts that are not in the current work requirement set
- **No emulation**

RDz – Project/Work Request Coverage



RDz – Integrated Lifecycle Coverage



Additional RAA/RDz Development Savings

▪ **Productivity Benefits:**

- ▶ Even for experienced TSO veterans, but especially for new PC-aware developers:
 - See (view) substantially more code
 - Integrated analyze/edit/compile/debug/upload/download – access to z/Resources
 - Instantaneous response time during develop and test
 - Function/Enter keys do not interrupt z/OS CPU
 - 40+ (over forty) discrete edit/test/debug productivity features available with RDz:
 - Not available using ISPF
 - And only a handful of ISPF features not available on RDz

▪ **Web Services**

- ▶ RDz provides wizards to automate service creation:
 - Bottom-up:
 - From a CICS program COMMAREA and an IMS IO-AREA
 - Top-down:
 - From a WSDL (RDz generates call-able, "wrapper" programs)
 - Both Bottom-up/Top-down
- ▶ This takes SOA development for traditional mainframe applications from the Stone Age to the 21st Century
 - Anecdotal evidence suggests that doing Web Service development using RDz provides several magnitudes of order productivity increases

What Benefits Will You Get With RAA?

■ Save costs for:

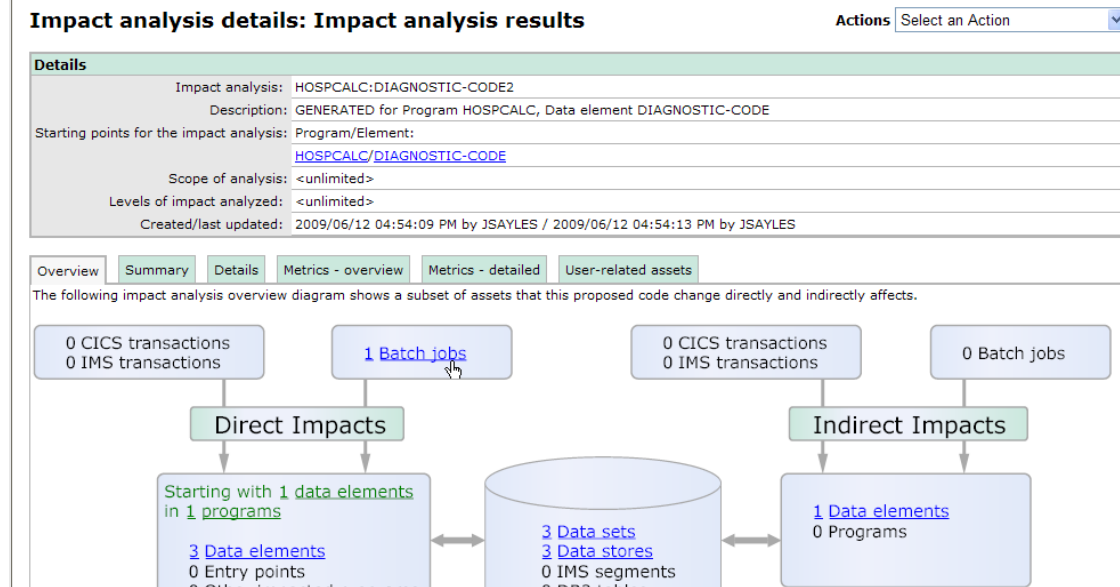
- ▶ Contracts and bids
- ▶ Project estimation
- ▶ Project management
- ▶ Analysis
- ▶ Code understanding
- ▶ Development
- ▶ Maintenance
- ▶ Testing and Bug-Fixes
- ▶ Documentation

■ Offload TSO/ISPF code analysis:

- ▶ TSO/ISPF 3.4 Dataset searches
- ▶ Edit session internal program "Find commands"
- ▶ File browsing
- ▶ Program compiles:
 - Listings/X-Refs, etc.

■ During project estimation/code analysis**

- ▶ MIPS savings from: **30% → 60%**
- ▶ Productivity increases for analyzing code from: **30% → 60%**



** Reported
by RAA
users



Three Kinds of Savings**

1. Mainframe compile savings:

- Batch compiles
- TSO/Foreground compiles
- Both include the CICS and DB2 pre-processors
- Compile cost reduction: **~80%**
- Compile productivity savings: **~25%**

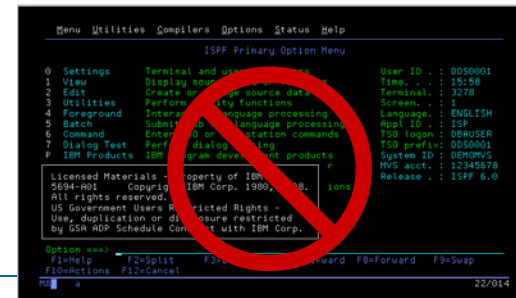
2. TSO/CPU Savings - Per user/Per session

- CPU Costs
- TSO Session Connect time
- DASD/Virtual Storage/Swap Space
- Combined savings: **35% → 75%**

3. Productivity gains

- Analysis (assuming RAA): **20% → 50%**
- Edit: **10% → 25%**
- Debug: **10% → 20%**

** Reductions based on RDz customer experiences



Mainframe Compile Savings

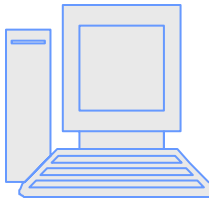
- **Reduced number of Compiles –**

- ▶ CPU cycles
- ▶ EXCP

- **RDz provides five levels of reduced compile costs:**

- ▶ **Content Assist**
 - Code completion – significantly reduces typos
- ▶ **Validation**
 - Interactive syntax checking – catches typos as they occur
- ▶ **Local Syntax Check**
 - Offloaded compile

-
- ▶ **Remote Compile**
 - z/OS compile without creating binaries – significantly less expensive than...
 - ▶ **Full compile**
 - Enterprise compiler



**Workstation
Based**



**Mainframe
Compile**

Mainframe TSO Session Savings

■ TSO/ISPF Development:

- ▶ Connect/Disconnect/Reconnect (and all ISPF Dataset Allocation/De-Allocation)
- ▶ Browse
- ▶ Edit
- ▶ Search (3.4)
- ▶ Dataset Management
- ▶ Job submission and review
- ▶ Unit testing – PD Tools, Xpeditor, etc.
- ▶ Integration testing

■ DB2/SQL Testing

- ▶ SPUFI/QMF:
 - SQL statement testing
 - Test data management

■ CICS Testing

■ IMS Testing

How are TSO savings possible?

```
Menu Utilities Compilers Options Status Help
ISPF Primary Option Menu

0 Settings Terminal and user parameters User ID : DDS0001
1 View Display source of files Time : 15:58
2 Edit Create or change files Terminal : 3278
3 Utilities Perform utility functions Screen : 1
4 Foreground Interactive language processing Language : ENGLISH
5 Batch Submit job for language processing Appl ID : ISP
6 Command Enter TSO or Workstation commands TSO logon : DBAUSER
7 Dialog Test Perform dialog test TSO prefix : DDS0001
P IBM Products IBM program development products System ID : DEMOMVS
MVS acct. : 12345678
Release : ISPF 6.0

Licensed Materials - Property of IBM
5694-A01 Copyright IBM Corp. 1980, 2008.
All rights reserved.
US Government Users Restricted Rights -
Use, duplication or disclosure restricted
by GSA ADP Schedule Contract with IBM Corp.
```

```
IBM* Licensed Materials - Property of IBM
(C) Copyright IBM Corp. 1982, 1992. All rights reserved.
US Government Users Restricted Rights - Use, duplication
or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
* Trademark of International Business Machines Corp.

QMF HOME PANEL
Version 3 Release 1.1

Query Management Facility

Connected to DB2PMVS

Type command on command line or use PF keys. For help, press PF1 or type HELP.
```

```
TERMINID: 0420 PRODUCTION CICS
Customer Assistance and Problem Reporting, call the Help desk at 301-405-7763.

CCCCC IIII CCCCC SSSSS
CCCCCCC IIII CCCCCC SSSSSSS
CCCC CC III CCCC CC SSSS SS
CCC III CCC SSSS
CCC III CCC SSSS
CCCC CC III CCCC CC SS SSSS
CCCCCCCC CCCCCC SSSSSSS
CCCCC CCCCC SSSSSS 4.1.0

Fill in your USERID and PASSWORD then press ENTER to sign on to CICS
USERID: _____ PASSWORD: _____ BYPASS INITIAL KEYWORD: _

PRESS: ENTER=Signon, F1=Help, F3=Exit CICS
```

Every ←Enter and every PF-Key pressed while in a TSO session results in CPU cycles – which end up as charge-back costs***



Key to Lowering TSO/Session CPU Costs

- For those development activities that don't require you to offload your z/OS development sub-system, use RAA/RDz and PD Tools to minimize TSO access and activity



The screenshot displays the z/OS development environment. On the left, a project tree shows a hierarchy of files including 'ZOSProject [Offline]', 'MVS Subproject [DEMOMVS.D]', and various COBOL files like 'HOSPCALC.cbl', 'HOSPEDIT.cbl', 'HOSPORT.CBL', 'HOSPRCH.CBL', 'IMSBatch.cbl', 'VARPGM.CBL', 'calc.cbl', 'conditional.cbl', 'hospin.dat', 'newProg.cbl', 'seqfileproct.cbl', and 'test4.cbl'. The main editor window shows COBOL code with a context menu open over the 'IF SUITE' statement. The menu options include 'Save', 'Cut', 'Copy', 'Paste', 'Select', 'Deselect', 'Filter view', 'Source', 'Refactor', 'Open Declaration', 'Open Perform Hierarchy', 'View', 'Run As', 'Debug As', 'Profile As', 'Validate', 'Team', 'Compare With', 'Replace With', 'Local Syntax Check', 'Browse Copy Member', 'Open Copy Member', 'Content assist', and 'Start Flagging Changed Lines'. The 'Refactor' option is highlighted, with a sub-menu showing 'Rename' and 'Remove Noise Words'. The bottom status bar shows 'Line Perform Statement 620'.

What Cost-Saving Numbers Do We Have?

- **Three kinds:**

- 1. IBM internal study of a "trivial code maintenance" project**

- ▶ Detailed SMF data
- ▶ Comparison between TSO/ISPF and RDz – for TSO CPU cycles (translated to Service Units and MIPS)
- ▶ What is important is the relative savings factor (% MIPS reduction) for this workload

- 2. Anecdotal data from customers**

- 3. Your data (and specific inputs)**

- ▶ The detailed part of this session
- ▶ You should have with you:
 - ▶ Cost per/developer
 - ▶ Cost per/batch compile
 - ▶ Cost per TSO Session
- ▶ With this, we can predict RDz MIP savings

IBM "Trivial Edit" Usage Pattern Benchmark – Scenario Details

RDz – 30 minute session	TSO/ISPF – 30 minute session
- Connect to RDz MOP system	Logon to TSO
- Create a remote project and subproject	Enter ISPF
- Edit a remote program	ISPF Edit Session
- Use content assist to add new COBOL and CICS statements.	Manually type statements
- Open a remote copy member	Split Screen – open and edit a copybook
- Put an error in the program	Edit (put in a syntax error on purpose)
- Local Syntax Check with Refresh dependencies, with 2 dependant files	Submit a batch compile – Split screen – go to Outlist, View results
- Edit the remote program and fix the error	Edit – fix the syntax error
- Local Syntax check again	Submit a batch compile – Split screen – go to Outlist, View results
- Edit the remote program to fix an other error	Edit – fix the other syntax error
- Local syntax check again → no more errors	Submit a batch compile – Split screen – go to Outlist, View results
- From the remote source program, generate the JCL to Compile and Link	Edit the Compile/Link/GO JCL
- Edit the JCL	ISPF Edit
- Submit the JCL	Submit
- Open the Job Monitor	Split Screen – go to Outlist
- View the Job Output – purge -- JCL error , dataset not found	View the results – Purge the JES Queue (JCL Error)
- Modify the program properties group to change the dataset name	Edit the JCL – change the DD card
- Submit the JCL	Submit the job
- Open the Job Monitor	Navigate to the Outlist facility
- View the Job Output -- flush -- JCL error, another dataset not found again	View the results – another JCL error – Purge the JES Queue (JCL Error)
- Edit and Submit the JCL again	Edit JCL – change DD card Submit the Job again
- Open the Job Monitor	Navigate to the Outlist facility
- View the Job Output – find that the JOB ended with RC=0	Browse – find the JOB ended with RC=0
- From the remote connection, open the Host Connection emulator to connect to CICS	Leave TSO – Logon to CICS Subsystem
- CADP to define the transaction to debug;	CADP to define the transaction to debug;
The 30 minutes window raised , disconnect from RDz	Logoff CICS

Savings:

5,620,300 Service Units/day ***

CPU Savings: 42%

***** Based on 10 developers, working 7 hours/day, using RDz instead of TSO/ISPF**

Details:

- **TSO Session:**
 - CPU = 191,842 SU/per user/per hour
- **RDz Development:**
 - CPU = 111,552 SU/per user/per hour
- **Savings: 80,290 Service Units/per user**

***** See Notes**

IBM "Trivial Edit" Usage Pattern Benchmark – Additional Benefits

- **Note that the previous benchmark and ROI calculations:**
 - ▶ **Are based on a "Worst Case" usage scenario for RDz**
 - ▶ **Are minus a number of additional returns on investment:**
 1. **RAA cost savings (substantial)**
 2. **PD Tools cost savings (substantial savings in the acquisition and license fees)**
 3. **Productivity Benefits:**
 - Assume **conservative** benefits broken out by lifecycle phase:

Developers time spent in:	Developer time	Productivity improvements
Analysis/Design/Understanding	35.00%	42.00%
Editing	20.00%	28.00%
Compile	5.00%	5.00%
Debug	15.00%	5.00%
Other	25.00%	0.00%
	100.00%	
Overall Productivity Increase		21.30%

- **What does that all add up to?**

Anecdotal Data From Customers

- **Large bank in Germany** did an even more precise and detailed study (as on the previous slides) but with a different use case.
 - ▶ **Benchmark results:**
 - **75% Percent CPU Savings**
 - **With 100 Developers - €3,600,000 savings/Annually** (\$4,160,000 U.S.)
- **Other European Banks benchmarks and feedback – With RDz:**
 - ▶ **Reduced TSO consumption by as much as 80%**
 - ▶ **Saved 2 Euros an hour/Per Developer** using RDz versus the mainframe
- **U.S. Telecomm company – With RDz:**
 - ▶ **Reduced costs by more than US\$75,000** on a single project
- **European Services Provider** – estimates that by rolling RDz out to 600 users will save ~ **€74,000,000 over five years** (\$102,490,000 U.S.)
- **Financial services provider in Sweden:**
 - Saves approximately **US \$1.6 million** on licensing costs & improved ABEND analyses, file alterations and application debugging processes when they replaced existing debugging software with PD Tools
 - Reduced 90% of time spent tuning programs and discovering performance bottlenecks when they implemented Application Performance Analyzer to help diagnose issues in IMS & DB2

Can we guarantee these MIPS reductions?

- **Sorry – but:**

- ▶ Yours might be more 😊
- ▶ Yours might be equal-to 😊
- ▶ Yours might be less 😞
- ▶ It depends on:

- **The kind of workflow and tools usage:**

- The more analysis/edit/compile/unit test - the bigger the benefits

- **Your developers' willingness to adopt a better development model:**

- **Old model:**

- Do some analysis (usually incomplete and cursory)
- Edit
- Compile
- Run the integration stream and verify results
- Repeat

- **The right – or Best Practices development model:**

- **Deep analysis** (offline – in less time and with vastly improved results with RDz or RAAi)
- **Edit/Compile using RDz** (offline)
 - Repeat (also offline)
- **Compile and integration test using RDz/PD Tools on the host** – far fewer test runs

How to Calculate Your Own Savings

- Time to work through the ROI spreadsheet

- Enterable fields are in:

- ▶ Blue
- ▶ Yellow

Legend

IBM Inputs

Customer Inputs

Calculated



- Gray fields are spreadsheet computations – based solely on:

- ▶ Your enter-able values
- ▶ That's it... no "secret sauce"

- The following slides describe the values and formulae in the ROI calculator



ROI Spreadsheet – 1 of 4 (Note – values not specific to any customer)

Number of Developers	Year 1	70	Cost of each License	\$5,253.00
	Year 2	0	Yearly cost of license PPA contract	\$1,050.60
	Year 3	0	Working days in a year	220
	Total	70	First year rate of return (<100%)	80.0%
Cost of one hour of mainframe CPU		\$913.00	This is a rate used to reduce the benefit/return calculated in the first year of use of the product: it accounts for lower productivity product adoption ramp up, possible problems, during slower speed to market this is an enter-able field. It is not	
Cost of Developer (yearly)		\$120,000.00		
(daily)		\$545.45		
Cost of 1 TSO per month		\$20,000.00		
Cost of 1 TSO per day per developer		\$16.00		
Cost of Compiles per day		\$26.40		
Number of compiles per day per developer		12		
Average Cost of 1 Compile		\$2.20		
Number of Developers needed in setup phase	Year 1	1.00		
	Year 2	0.00		
	Year 3	0.00		
Number of Developers needed each year for admin/mgmt		0.00		

Notes:

- ▶ IBM Sales Team will give you the: Cost of each license and Yearly cost of license PPA contract
- ▶ These are conservative TSO costs (in this screen capture) – based on world-wide corporate statistics
- ▶ Your values for:
 - Developer costs – fully loaded
 - TSO costs/Compile costs
 - Setup and administration conservative number



ROI Spreadsheet – 2 of 4

Cost of Initial Services

Year 1	\$50,000.00	14 days	days per developer	2
Year 2	\$0.00	0 days	developers per training	10
Year 3	\$0.00	0 days	Cost of one day trainer	0.00
	\$50,000.00	14		

Direct Support

Year 1	0.00
Year 2	0.50
Year 3	0.50

HW Growth

Year 1	\$0.00	Additional HW cost per developer	\$0.00
Year 2	\$0.00		
Year 3	\$0.00		

Developers time spent in:

	Developer time	Productivity improvements	
Analysis/Design/Understanding	35.00%	42.00%	Average reduced num. of compiles
Editing	20.00%	28.00%	80%
Compile	5.00%	5.00%	
Debug	15.00%	5.00%	
Other	25.00%	0.00%	
	100.00%		
Overall Productivity Increase		21.30%	Average TSO sessions reduction
			40%

- ▶ **Education can be zeroed out, if choosing RDz Distance Learning**
- ▶ **Productivity improvements are based on:**
 - Customer feedback
 - IBM's laboratory research (measured against green-screen development tools)
- ▶ **Percentages for reduced num. of compiles, Average TSO sessions reduction and productivity are conservative**
- ▶ **For hardware, assume that you need a fast processor, 2 GIGS/RAM/10 GIGS of free disk space**



ROI Spreadsheet – 3 of 4

Rollout	70	0	0			70
Costs						
Software	\$367,710	\$0	\$0			\$367,710
Subscription		\$73,542	\$73,542	\$73,542		\$220,626
Education	\$50,000	\$0	\$0			\$50,000
Setup/Admin Staff	\$120,000	\$0	\$0	\$0	\$0	\$120,000
Support Staff	\$0	\$60,000	\$60,000	\$60,000	\$60,000	\$240,000
Hardware	\$0	\$0	\$0			\$0
Total Costs	\$537,710	\$133,542	\$133,542	\$133,542	\$60,000	<u>\$998,336</u>
Justification						
Compile CPU Savings	\$325,248	\$325,248	\$325,248	\$325,248	\$325,248	\$1,626,240
TSO CPU Savings	\$98,560	\$98,560	\$98,560	\$98,560	\$98,560	\$492,800
Productivity	\$1,431,360	\$1,789,200	\$1,789,200	\$1,789,200	\$1,789,200	\$8,588,160
Total Justification	\$1,855,168	\$2,213,008	\$2,213,008	\$2,213,008	\$2,213,008	<u>\$10,707,200</u>
Return	\$1,317,458	\$2,079,466	\$2,079,466	\$2,079,466	\$2,153,008	\$9,708,864

Notes:

- ▶ These are just #s based on values in previous slide
 - Yours will be entirely different
- ▶ Note the zeros for:
 - On-site education
 - This does not mean you don't need education – it means you will be taking it via remote, Distance Learning
 - Hardware upgrades
- ▶ Recall – very conservative TSO CPU Savings justification

Questions



Thank You

© Copyright IBM Corporation 2009. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. IBM, the IBM logo, Rational, the Rational logo, Telelogic, the Telelogic logo, and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.





IBM Software Group

Appendices:
- CPU Interrupt data points

A horizontal decorative bar at the bottom of the slide contains a series of small, square icons. From left to right, the icons include: a green square, a yellow square, a red square, a purple square, a cyan square, a grayscale image of a highway, a circular arrow icon, a grayscale image of a woman's face, a grayscale image of a hand, and several grayscale abstract shapes.

Rational. software

TSO/CPU Costs – Specific Chargeback (Partial) List – 1 of 7

- Every ←Enter and every PF-Key pressed while in a TSO session results in a CPU interrupt and charge-back

- This includes (but is not limited to):
 - ▶ **Logging in/Logging out of TSO – ISPPROC Dataset Allocate/De-Allocate**
 - ▶ **ISPF Navigation**
 - Going anywhere (all ISPF and custom options) =**whatever**
 - ▶ **Issuing REXX and CLIST commands**
 - ▶ **Accessing Help (PF1)**
 - ▶ **Accessing ISPF Option 0 customization**
 - ▶ **Submitting Jobs**
 - ▶ **Managing Jobs:**
 - **Viewing Job results in TSO:**
 - Opening the JES logs
 - Browsing job results:
 - PF7/PF8/PF10/PF11 within the file
 - HEX ON/HEX OFF
 - Purging the JES queue
 - Closing the files

- **Option 1 – or "browse" function in 3.4, Outlist, SPUFI, etc.**
 - ▶ **Opening each file**
 - ▶ **PF2**
 - ▶ **PF5**
 - ▶ **PF7**
 - ▶ **PF8**
 - ▶ **PF9**
 - ▶ **PF10**
 - ▶ **PF11**
 - ▶ **All ISPF Primary Option commands**
 - **Ex.**
 - **Find**
 - **HEX (on/off)**
 - **Etc.**
 - ▶ **Closing the files**

- **Option 2 – or Edit function of 3.4**

(Partial list – common ISPF Edit actions):

- ▶ **Opening each file**

- ▶ **PF1**

- ▶ **PF2**

- ▶ **PF3**

- ▶ **PF4**

- ▶ **PF5**

- ▶ **PF6**

- ▶ **PF7**

- ▶ **PF8**

- ▶ **PF9**

- ▶ **PF10**

- ▶ **PF11**

- ▶ **Line commands**

- **Copy/paste/move/insert/delete/shift/eXclude**

- **Tabs**

- **Columns**

- **Text/Split**

Option 2 (Partial list – continued):

- **Primary commands**
 - ▶ **Reset**
 - ▶ **Find**
 - ▶ **Change**
 - All permutations
 - ▶ **Copy in an external PDS member**
 - ▶ **Create an external PDS member**
 - ▶ **HEX On/Off**
 - ▶ **Locate**
 - ▶ **Nulls**
 - ▶ **RES**
 - ▶ **NUM ON/OFF**
 - ▶ **Other commands...**
- **Save/Close the file**
- **Custom ISPF edit macros**

- **Dataset Management**

- ▶ **Allocate datasets**

- ▶ **Copy datasets**

- ▶ **Delete datasets**

- ▶ **Create Like**

- ▶ **Work with PDS/PDSE datasets**

- **Copy members**

- **Compress**

- **Browse PDS members – especially source files in 3.4**

- ▶ **Find statistics on datasets**

- **Working with DB2 SPUFI/QMF**

- ▶ **Navigation**

- ▶ **TSO/DB2 connect (IKJEFT01)**

- ▶ **The SPUFI and QMF dialogs**

- **Especially QMF**

- (Forget about QBE)

- ▶ **SQL statement code/test:**

- **All editing costs from the Edit slides**

- ▶ **Test data man:**

- **Insert**
 - **Update**
 - **Delete**

- ▶ **All browse operations - PF7/PF8/PF10/PF11/PF2/PF9**

- ▶ **Test table maintenance:**

- **DDL**
 - **DCL**

- **CICS Testing**

- ▶ **All Transaction execution**

- **Unit Testing with Xpeditor -- versus PD Tools and RDz**
 - ▶ **Running the Debugging tool – Costs are same as on the host**
 - ▶ **Running the Program – Costs are same as on the host**

- **Debugging the program - every mainframe debug session is a series of CPU interrupts - caused by PF Keys and <Enter>.**

- **All source file navigation that is not an action passed up to PD Tools is a CPU cost saving:**
 - ▶ **Browsing/Scrolling the file**
 - PF7/PF8
 - PF9/PF10
 - ▶ **Find**
 - ▶ **Opening an edit session in another window and modifying the source**
 - ▶ **Etc.**