Benchmarking z/OS Development Tasks -
Comparing Programmer Productivity using RDz and ISPF

Jon Sayles
RDz Technical Enablement
jsayles@us.ibm.com

Agenda and Disclaimer

- The Hypothesis
- Benchmark Methodology
  - Scenarios
- Benchmark Results
- Mitigating Factors

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

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

The Hypothesis

- For many decades it has been assumed that graphical development tools offer benefits over character-based technologies
- Concerns:
  - Relatively little in the way of fully-documented apples-to-apples comparison research
  - What exists is generally:
    - Well over a decade old
    - Research that is focused on:
      - Traditional Data Entry screens
      - SLOC (Source Lines of Code)
  - For z/OS Traditional Maintenance activities (COBOL, PL/I, HLASM, etc.) – SLOC is no longer **the** relevant productivity metric, as the primary usage model of these applications are:
    - Maintenance
    - Support

The IDE Efficiency Benchmarks

- In Q1 2010 IBM/Rational was asked to develop a series of Benchmarking Scenarios to measure IDE efficiency – for a standard z/OS Maintenance and Application Support work-load

- Specifically
  - Measure differences in task completion between
    - **IBM Product "A" –** ISPF 6.0 running on a z/10 processor
    - **IBM Product "B" –** Rational Developer for System z 7.6

- The entire set of Benchmark scenarios is a work-in-progress, however we have finished an "apples to apples" use case and finished measuring results with z/OS practitioners from varying backgrounds
  - Eighteen participants:
    - Average ISPF experience: **12.1 years**
    - Average RDz experience: **1.3 years**

- We are hoping to execute a full series of open-ended Benchmarks before the end of the year - which will factor in additional products:
  - SCLM
  - Static Analysis tools

Design of the Apples-to-Apples Scenario

- 100 separate ISPF-based typical z/OS maintenance and support programmer tasks (scripts available on request)
- Transcribed each ISPF task to the equivalent RDz development technique:
  - Note that the direction was: *Start from an existing ISPF set of tasks → convert to RDz-based workflow*
- As far as possible, attempts were made to remove "Human Factors":
  - Close-ended "click-for-click" instructions were created to minimize:
    - Differences in think/reaction time
      - "Press PF8 20 times" – "Press PgUp 20 times", etc.
    - Differences in Product experience
    - Differences in business application development experience
  - Detailed testing methodology instructions were sent to
  - Project participants were told that they were trying to find gaps between RDz and ISPF functionality
  - 50% of those that did both the RDz and ISPF scripts did the RDz scripts first – to mitigate "learning and retention" bias
- Caveat: This does not mean that the Benchmark results should be construed as Underwriter's Laboratories research.

Apples-to-Apples Benchmark Scripts

100 specific tasks documented in a detailed script, and broken into seven sub-categories:

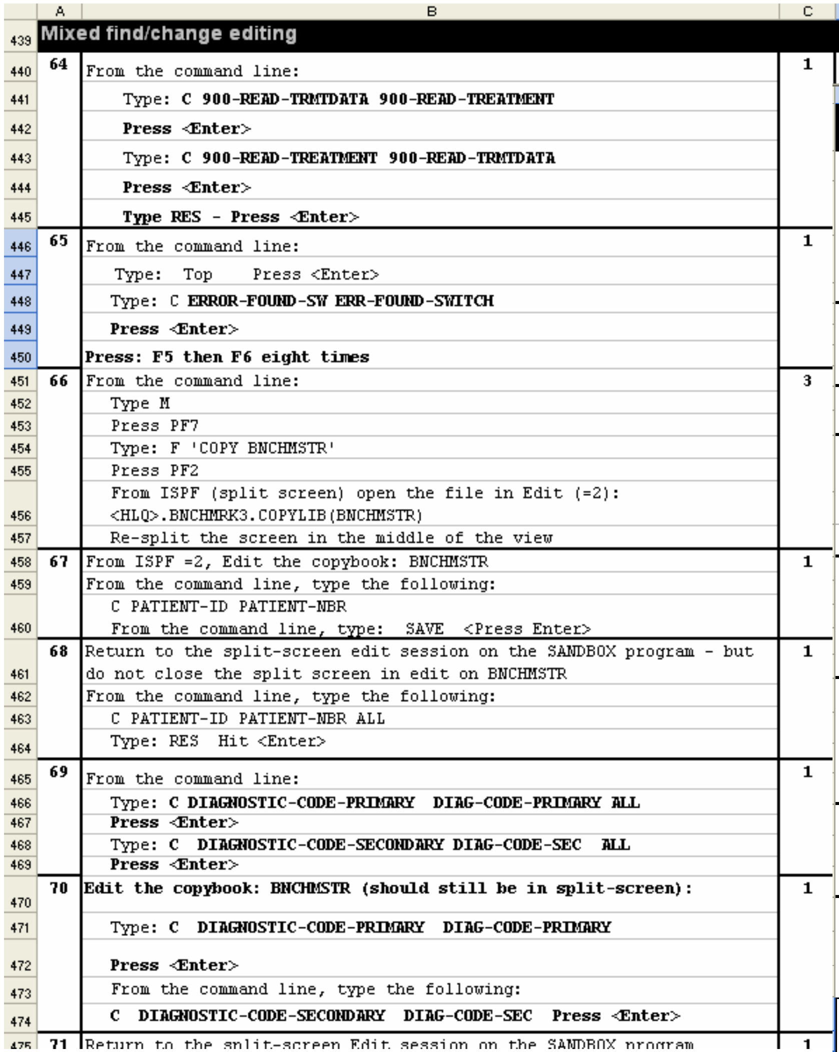
1. ISPF Source navigation
2. Program analysis (essentially, standard maintenance "Data Flow Analysis")
3. ISPF Editing operations (basically, the core ISPF Edit (=2) functionality)
4. COBOL statement coding
5. Syntax check/Syntax removal
6. Program compile & link
7. DB2/SQL work (test data manipulation and SQL statement create/test)

- We refined and vetted all tasks and workflow proportions in the scripts:
  - With ISPF and business application programming experts in IBM
  - With external business partners
  - With several customers under NDA

- We would be happy to eMail you the complete list of tasks and steps documented in the scripts:
  - If interested, please send a note to: Jon Sayles: jsayles@us.ibm.com

Apples-to-Apples Test Scripts

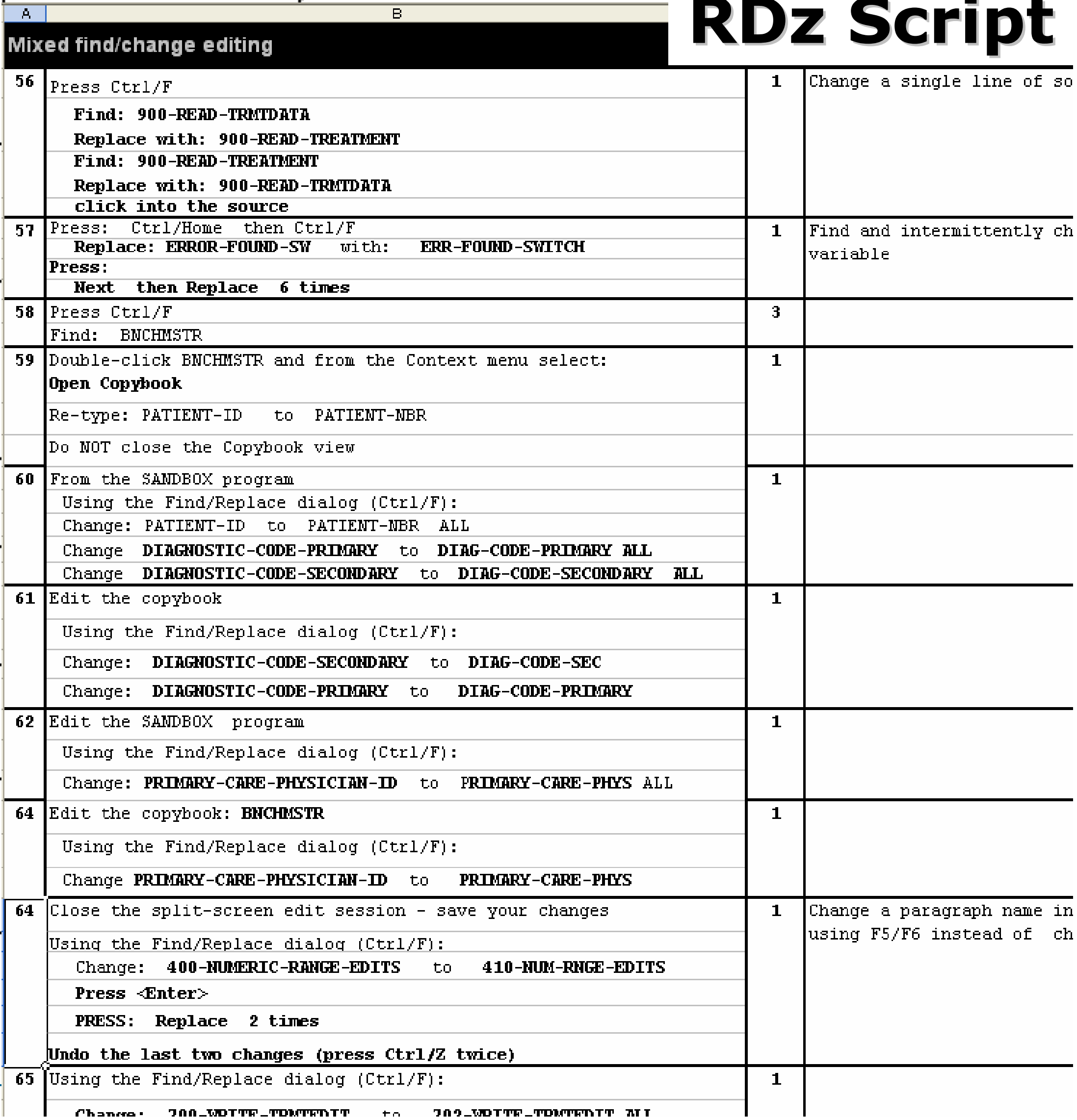
- The scripts were detailed to the PF-Key pressed, and mouse-click



A	B	C	D
Mixed find/change editing			
64	From the command line: Type: **C 900-READ-TRMTDATA 900-READ-TREATMENT** **Press <Enter>** Type: **C 900-READ-TREATMENT 900-READ-TRMTDATA** **Press <Enter>** **Type RES - Press <Enter>**	1	Change a single line of source
65	From the command line: Type: Top Press <Enter> Type: C **ERROR-FOUND-SW ERR-FOUND-SWITCH** **Press <Enter>** **Press: F5 then F6 eight times**	1	
66	From the command line: Type M Press PF7 Type: F 'COPY BNCHMSTR' Press PF2 From ISPF (split screen) open the file in Edit (=2): <HLQ>.BNCHMRK3.COPYLIB(BNCHMSTR) Re-split the screen in the middle of the view	3	
67	From ISPF =2, Edit the copybook: BNCHMSTR From the command line, type the following: C PATIENT-ID PATIENT-NBR From the command line, type: SAVE <Press Enter>	1	
68	Return to the split-screen edit session on the SANDBOX program - but do not close the split screen in edit on BNCHMSTR From the command line, type the following: C PATIENT-ID PATIENT-NBR ALL Type: RES Hit <Enter>	1	
69	From the command line: Type: C **DIAGNOSTIC-CODE-PRIMARY DIAG-CODE-PRIMARY ALL** **Press <Enter>** Type: C **DIAGNOSTIC-CODE-SECONDARY DIAG-CODE-SEC ALL** **Press <Enter>**	1	
70	**Edit the copybook: BNCHMSTR (should still be in split-screen):** Type: C **DIAGNOSTIC-CODE-PRIMARY DIAG-CODE-PRIMARY** **Press <Enter>** From the command line, type the following: C **DIAGNOSTIC-CODE-SECONDARY DIAG-CODE-SEC Press <Enter>**	1	
71	Return to the split-screen Edit session on the SANDBOX program	1	

ISPF Script

RDz Script



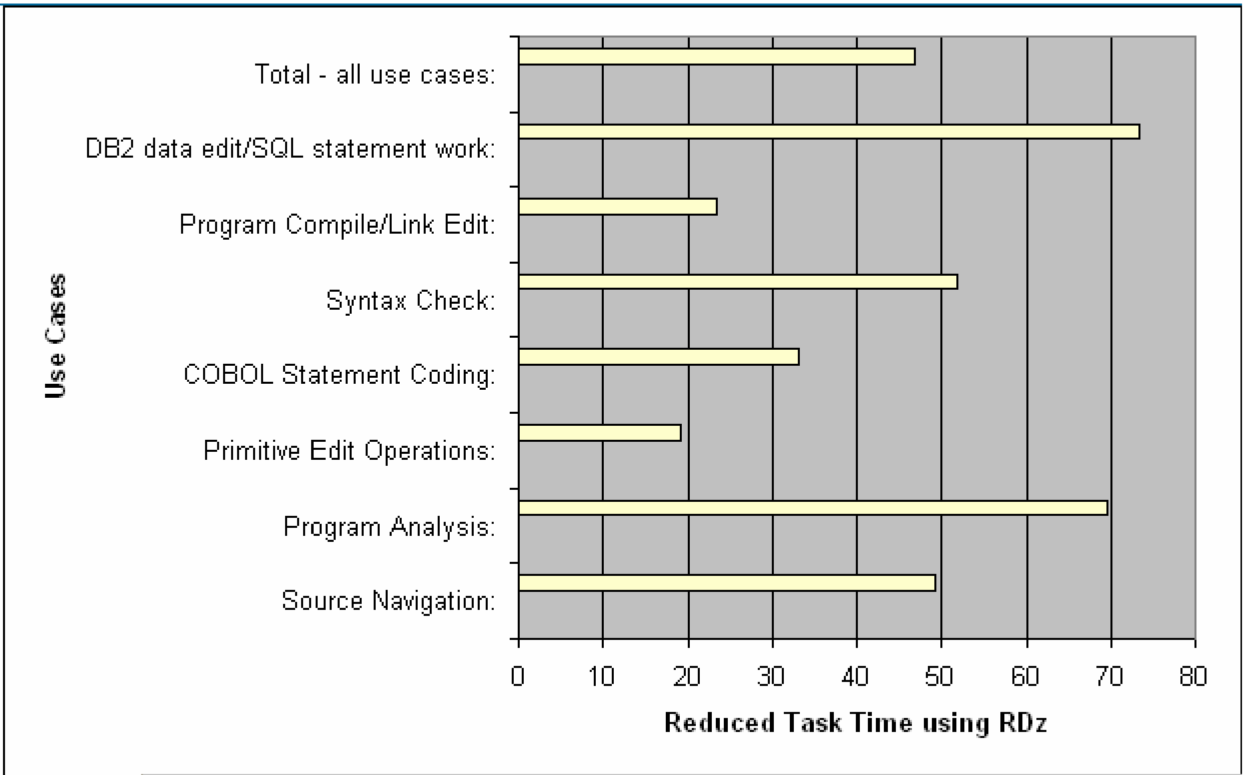
A	B		
Mixed find/change editing			
56	Press Ctrl/F **Find: 900-READ-TRMTDATA** **Replace with: 900-READ-TREATMENT** **Find: 900-READ-TREATMENT** **Replace with: 900-READ-TRMTDATA** **click into the source**	1	Change a single line of so
57	Press: Ctrl/Home then Ctrl/F **Replace: ERROR-FOUND-SW** with: **ERR-FOUND-SWITCH** **Press:** **Next then Replace 6 times**	1	Find and intermittently ch variable
58	Press Ctrl/F Find: BNCHMSTR	3	
59	Double-click BNCHMSTR and from the Context menu select: **Open Copybook** Re-type: PATIENT-ID to PATIENT-NBR Do NOT close the Copybook view	1	
60	From the SANDBOX program Using the Find/Replace dialog (Ctrl/F): Change: PATIENT-ID to PATIENT-NBR ALL Change **DIAGNOSTIC-CODE-PRIMARY** to **DIAG-CODE-PRIMARY ALL** Change **DIAGNOSTIC-CODE-SECONDARY** to **DIAG-CODE-SECONDARY ALL**	1	
61	Edit the copybook Using the Find/Replace dialog (Ctrl/F): Change: **DIAGNOSTIC-CODE-SECONDARY** to **DIAG-CODE-SEC** Change: **DIAGNOSTIC-CODE-PRIMARY** to **DIAG-CODE-PRIMARY**	1	
62	Edit the SANDBOX program Using the Find/Replace dialog (Ctrl/F): Change: **PRIMARY-CARE-PHYSICIAN-ID** to **PRIMARY-CARE-PHYS** ALL	1	
64	Edit the copybook: **BNCHMSTR** Using the Find/Replace dialog (Ctrl/F): Change **PRIMARY-CARE-PHYSICIAN-ID** to **PRIMARY-CARE-PHYS**	1	
64	Close the split-screen edit session - save your changes Using the Find/Replace dialog (Ctrl/F): Change: **400-NUMERIC-RANGE-EDITS** to **410-NUM-RNGE-EDITS** **Press <Enter>** **PRESS: Replace 2 times** **Undo the last two changes (press Ctrl/Z twice)**	1	Change a paragraph name in using F5/F6 instead of ch
65	Using the Find/Replace dialog (Ctrl/F): **Change: 200-WRITE-TRMTEDIT to 202-WRITE-TRMTEDIT ALL**	1	

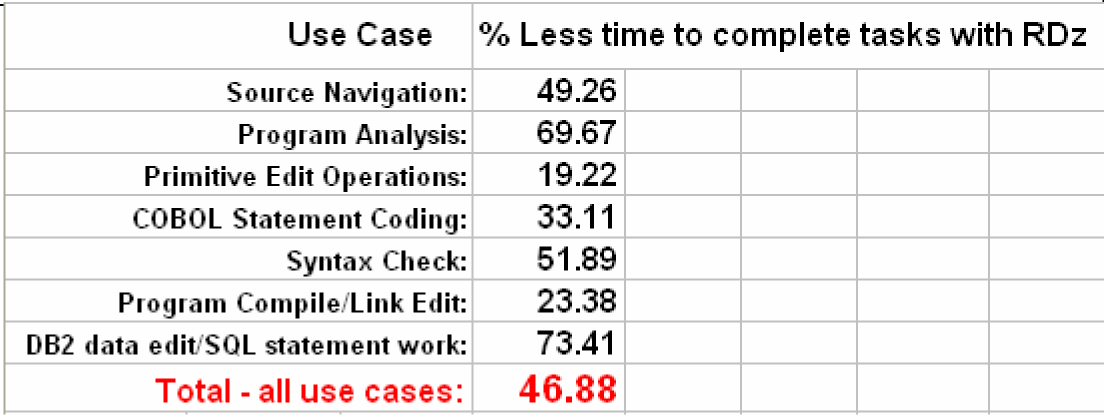
Task Summary Results – All Participant Subset

All participants ➔

- **ISPF veterans**
- **"New to ISPF" developers**

Note that **0** represents the ISPF baseline





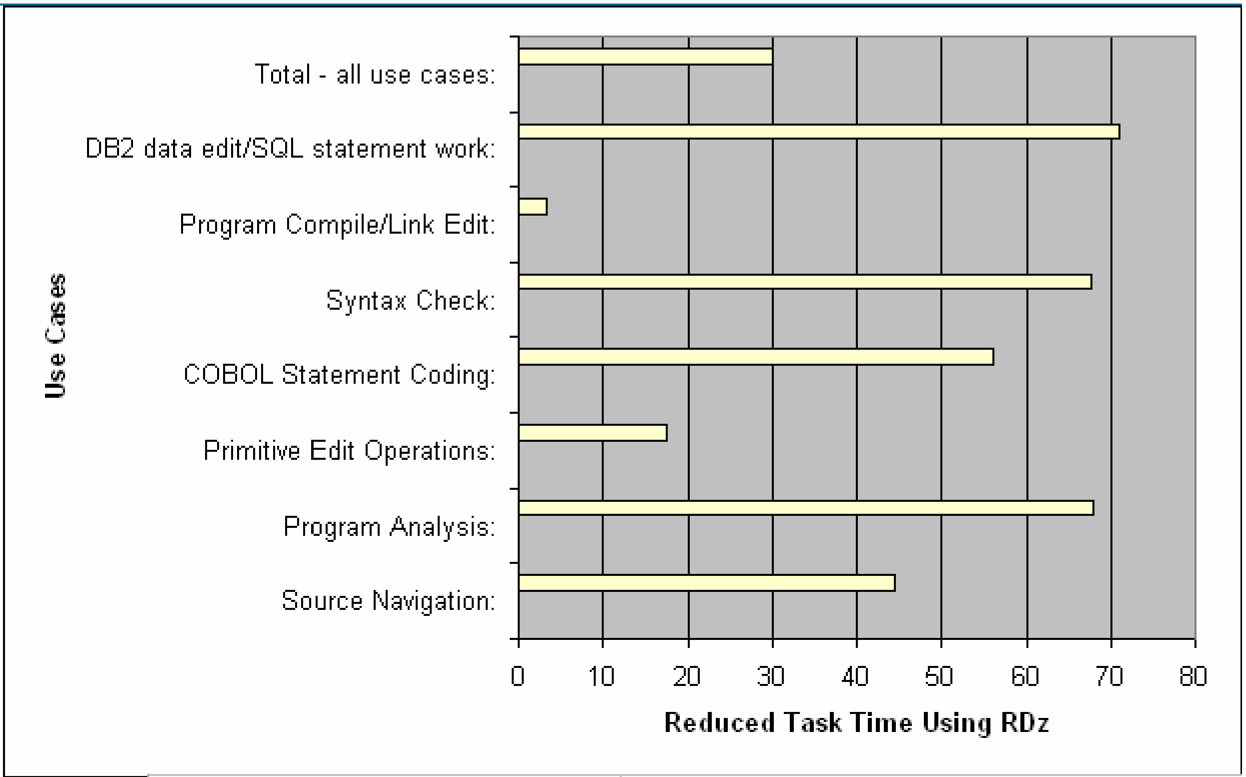
Use Case	% Less time to complete tasks with RDz
Source Navigation:	49.26
Program Analysis:	69.67
Primitive Edit Operations:	19.22
COBOL Statement Coding:	33.11
Syntax Check:	51.89
Program Compile/Link Edit:	23.38
DB2 data edit/SQL statement work:	73.41
Total - all use cases:	46.88

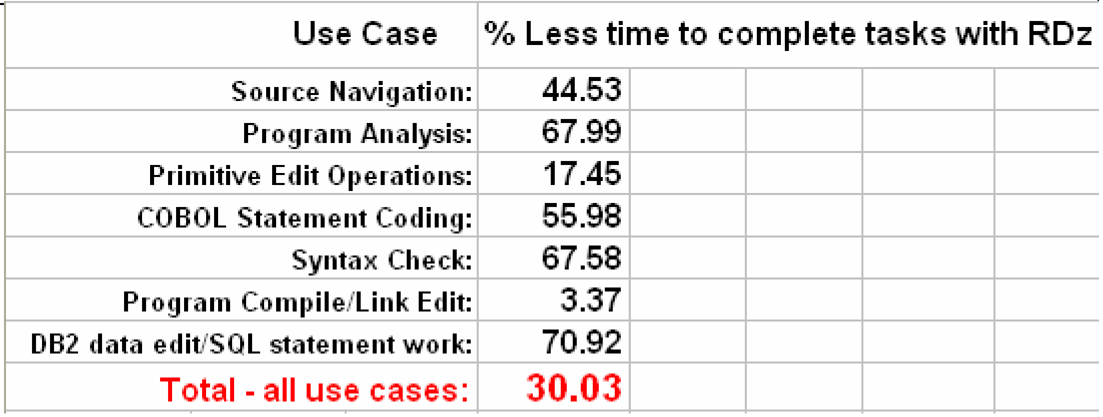
Task Summary Results – TSO "Top Gun" Participant Subset

Only participants with:

- Over 15 years of ISPF experience
- Recent ISPF work

Note that **0** represents the ISPF baseline





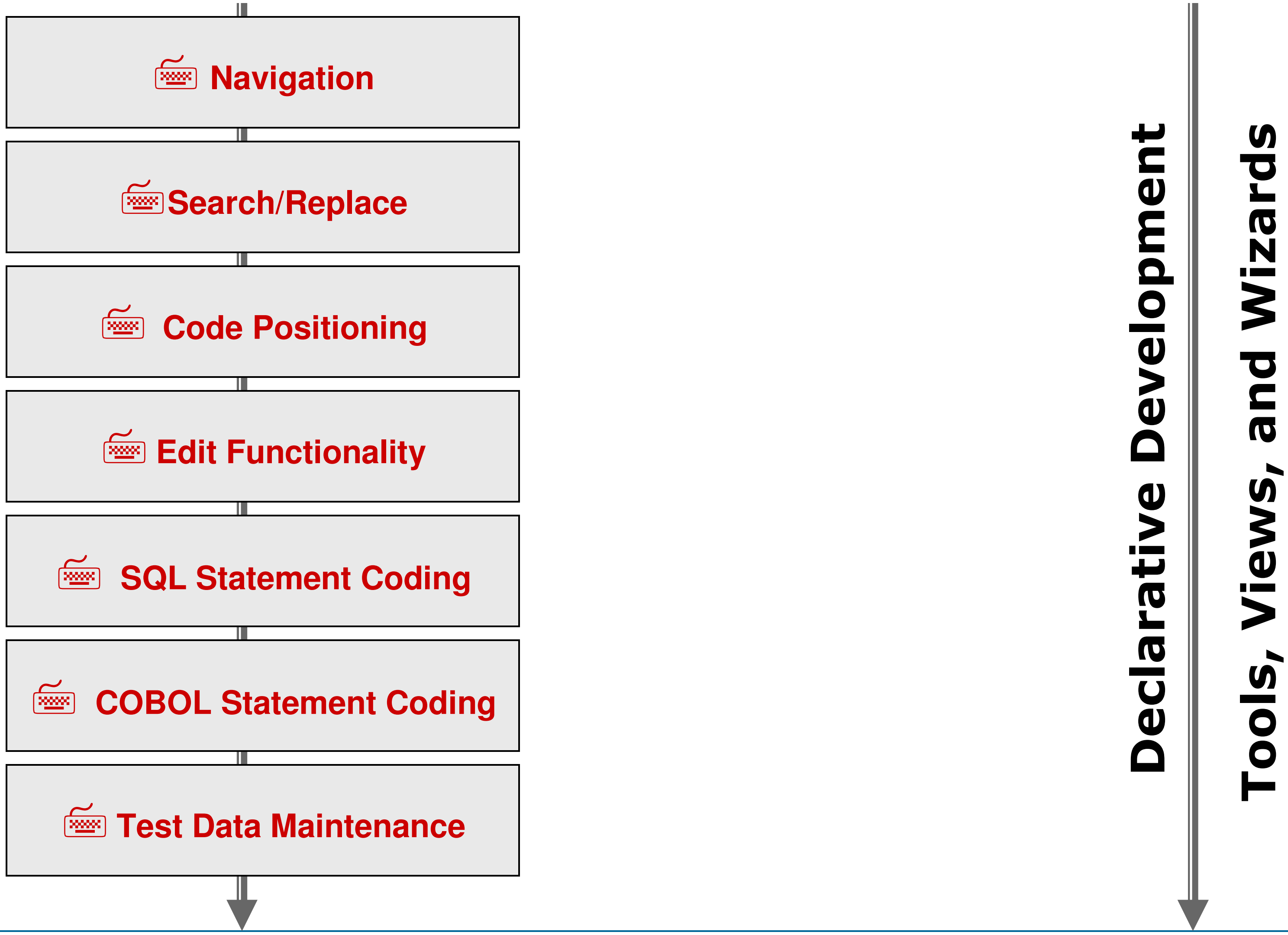
Use Case	% Less time to complete tasks with RDz
Source Navigation:	44.53
Program Analysis:	67.99
Primitive Edit Operations:	17.45
COBOL Statement Coding:	55.98
Syntax Check:	67.58
Program Compile/Link Edit:	3.37
DB2 data edit/SQL statement work:	70.92
Total - all use cases:	30.03

Analysis – and Feedback From Participants

- Four sources of productivity: **1. (Significantly) less typing with RDz**

Using ISPF – Typing for:

Using RDz



Navigation

Search/Replace

Code Positioning

Edit Functionality

SQL Statement Coding

COBOL Statement Coding

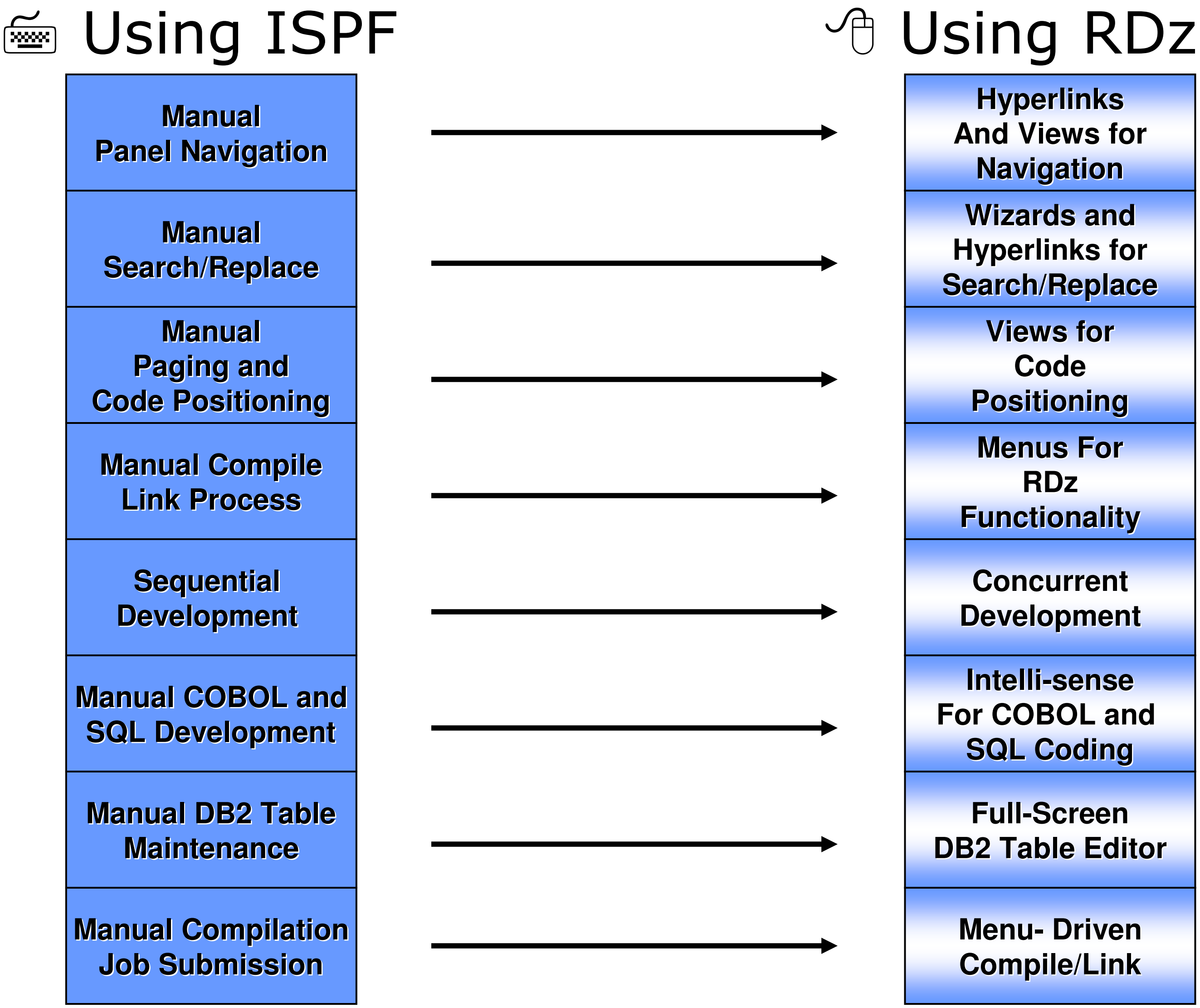
Test Data Maintenance

Declarative Development

Tools, Views, and Wizards

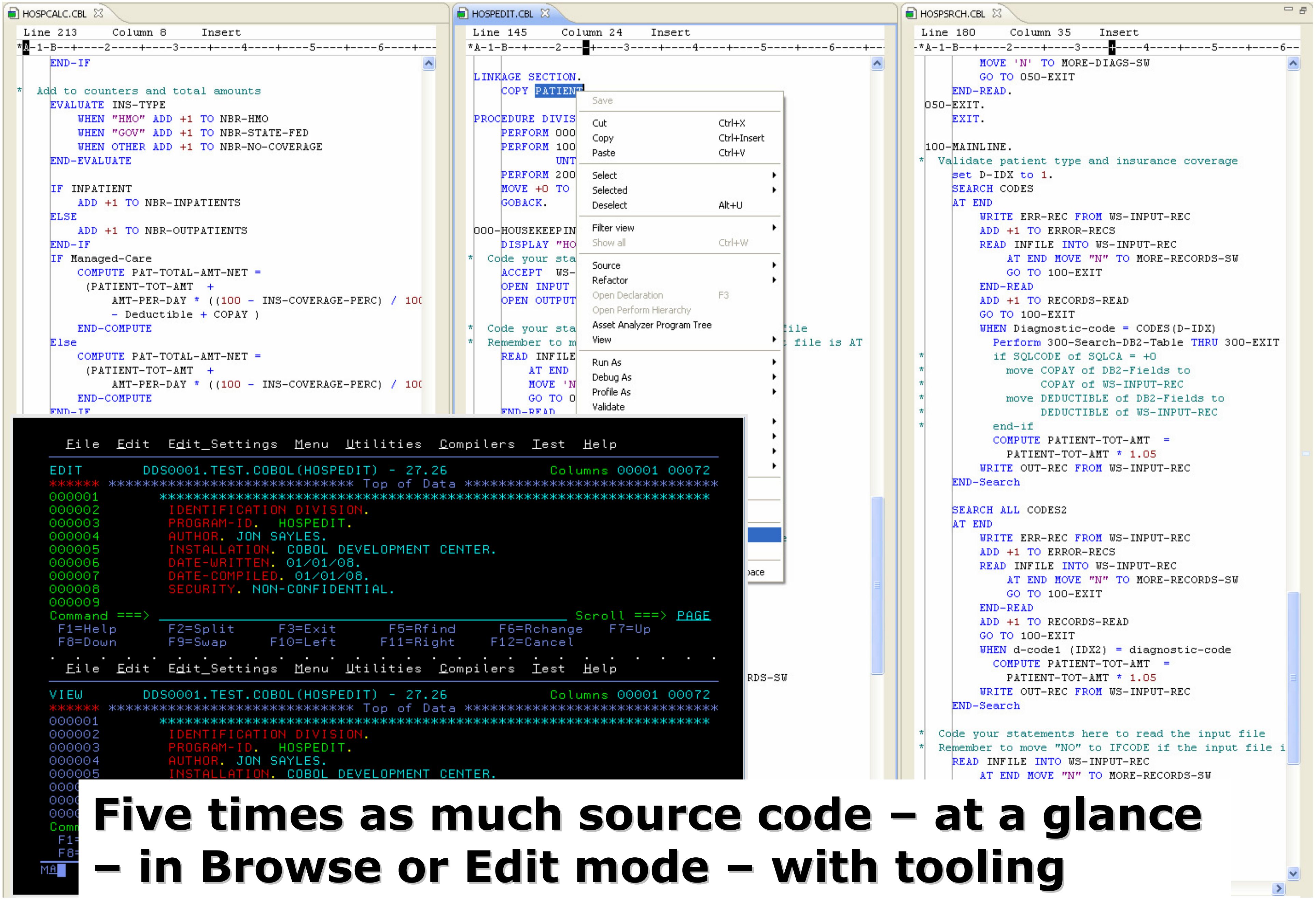
Analysis – and Feedback From Participants

- Four sources of productivity: **2. RDz Advanced Tooling**



Using ISPF	Using RDz
Manual Panel Navigation	Hyperlinks And Views for Navigation
Manual Search/Replace	Wizards and Hyperlinks for Search/Replace
Manual Paging and Code Positioning	Views for Code Positioning
Manual Compile Link Process	Menus For RDz Functionality
Sequential Development	Concurrent Development
Manual COBOL and SQL Development	Intelli-sense For COBOL and SQL Coding
Manual DB2 Table Maintenance	Full-Screen DB2 Table Editor
Manual Compilation Job Submission	Menu- Driven Compile/Link

Analysis – and Feedback From Participants - 3. Better use of Screen Real Estate



```
Line 213      Column 8      Insert
*A-1-B--+----2----+----3----+----4----+----5----+----6----+---
    END-IF

*  Add to counters and total amounts
    EVALUATE INS-TYPE
        WHEN "HMO" ADD +1 TO NBR-HMO
        WHEN "GOV" ADD +1 TO NBR-STATE-FED
        WHEN OTHER ADD +1 TO NBR-NO-COVERAGE
    END-EVALUATE

    IF INPATIENT
        ADD +1 TO NBR-INPATIENTS
    ELSE
        ADD +1 TO NBR-OUTPATIENTS
    END-IF
    IF Managed-Care
        COMPUTE PAT-TOTAL-AMT-NET =
         (PATIENT-TOT-AMT  +
            AMT-PER-DAY * ((100 - INS-COVERAGE-PERC) / 100
            - Deductible + COPAY )
        END-COMPUTE
    Else
        COMPUTE PAT-TOTAL-AMT-NET =
         (PATIENT-TOT-AMT  +
            AMT-PER-DAY * ((100 - INS-COVERAGE-PERC) / 100
        END-COMPUTE
    END-IF
```

```
Line 145      Column 24      Insert
*A-1-B--+----2---+----3----+----4----+----5----+----6----+--

 LINKAGE SECTION.
     COPY PATIENT

 PROCEDURE DIVIS
     PERFORM 000
     PERFORM 100
             UNT
     PERFORM 200
     MOVE +0 TO
     GOBACK.

 000-HOUSEKEEPIN
     DISPLAY "HO
*    Code your sta
     ACCEPT  WS-
     OPEN INPUT
     OPEN OUTPUT

*    Code your sta                 file
*    Remember to m               file is AT
     READ INFILE
         AT END
         MOVE 'N
         GO TO 0
     END-READ

                                   RDS-SW
```

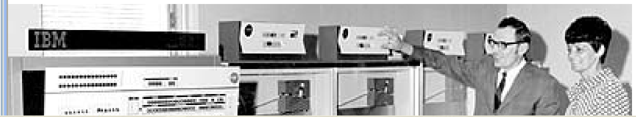
```
Line 180      Column 35      Insert
-*A-1-B--+----2----+----3----+----4----+----5----+----6--
        MOVE 'N' TO MORE-DIAGS-SW
        GO TO 050-EXIT
    END-READ.
 050-EXIT.
    EXIT.

 100-MAINLINE.
*  Validate patient type and insurance coverage
    set D-IDX to 1.
    SEARCH CODES
    AT END
        WRITE ERR-REC FROM WS-INPUT-REC
        ADD +1 TO ERROR-RECS
        READ INFILE INTO WS-INPUT-REC
            AT END MOVE "N" TO MORE-RECORDS-SW
            GO TO 100-EXIT
        END-READ
        ADD +1 TO RECORDS-READ
        GO TO 100-EXIT
        WHEN Diagnostic-code = CODES(D-IDX)
          Perform 300-Search-DB2-Table THRU 300-EXIT
*         if SQLCODE of SQLCA = +0
*           move COPAY of DB2-Fields to
*                COPAY of WS-INPUT-REC
*           move DEDUCTIBLE of DB2-Fields to
*                DEDUCTIBLE of WS-INPUT-REC
*         end-if
          COMPUTE PATIENT-TOT-AMT  =
            PATIENT-TOT-AMT * 1.05
        WRITE OUT-REC FROM WS-INPUT-REC
    END-Search

    SEARCH ALL CODES2
    AT END
        WRITE ERR-REC FROM WS-INPUT-REC
        ADD +1 TO ERROR-RECS
        READ INFILE INTO WS-INPUT-REC
            AT END MOVE "N" TO MORE-RECORDS-SW
            GO TO 100-EXIT
        END-READ
        ADD +1 TO RECORDS-READ
        GO TO 100-EXIT
        WHEN d-code1 (IDX2) = diagnostic-code
          COMPUTE PATIENT-TOT-AMT  =
            PATIENT-TOT-AMT * 1.05
        WRITE OUT-REC FROM WS-INPUT-REC
    END-Search

*  Code your statements here to read the input file
*  Remember to move "NO" to IFCODE if the input file i
    READ INFILE INTO WS-INPUT-REC
        AT END MOVE "N" TO MORE-RECORDS-SW
```

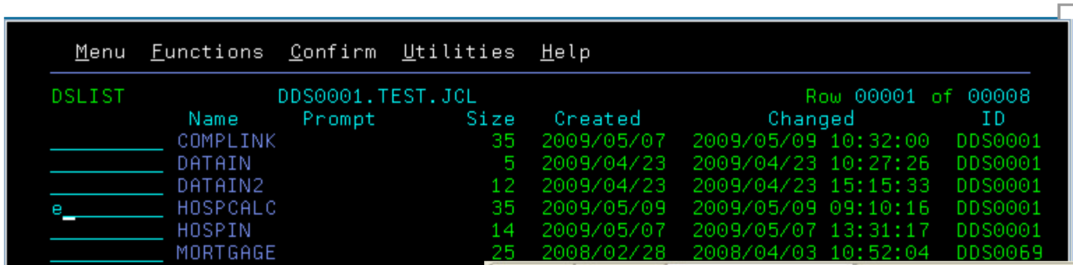
```
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
 ---------------------------------------------------------------------
 EDIT       DDS0001.TEST.COBOL(HOSPEDIT) - 27.26          Columns 00001 00072
 ****** ***************************** Top of Data ******************************
               ************************************************************
               IDENTIFICATION DIVISION.
               PROGRAM-ID.  HOSPEDIT.
               AUTHOR. JON SAYLES.
               INSTALLATION. COBOL DEVELOPMENT CENTER.
               DATE-WRITTEN. 01/01/08.
               DATE-COMPILED. 01/01/08.
               SECURITY. NON-CONFIDENTIAL.

 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
 ---------------------------------------------------------------------
 VIEW       DDS0001.TEST.COBOL(HOSPEDIT) - 27.26          Columns 00001 00072
 ****** ***************************** Top of Data ******************************
               ************************************************************
               IDENTIFICATION DIVISION.
               PROGRAM-ID.  HOSPEDIT.
               AUTHOR. JON SAYLES.
               INSTALLATION. COBOL DEVELOPMENT CENTER.



 Comm
  F1=
  F8=
 MA
```

Five times as much source code – at a glance – in Browse or Edit mode – with tooling

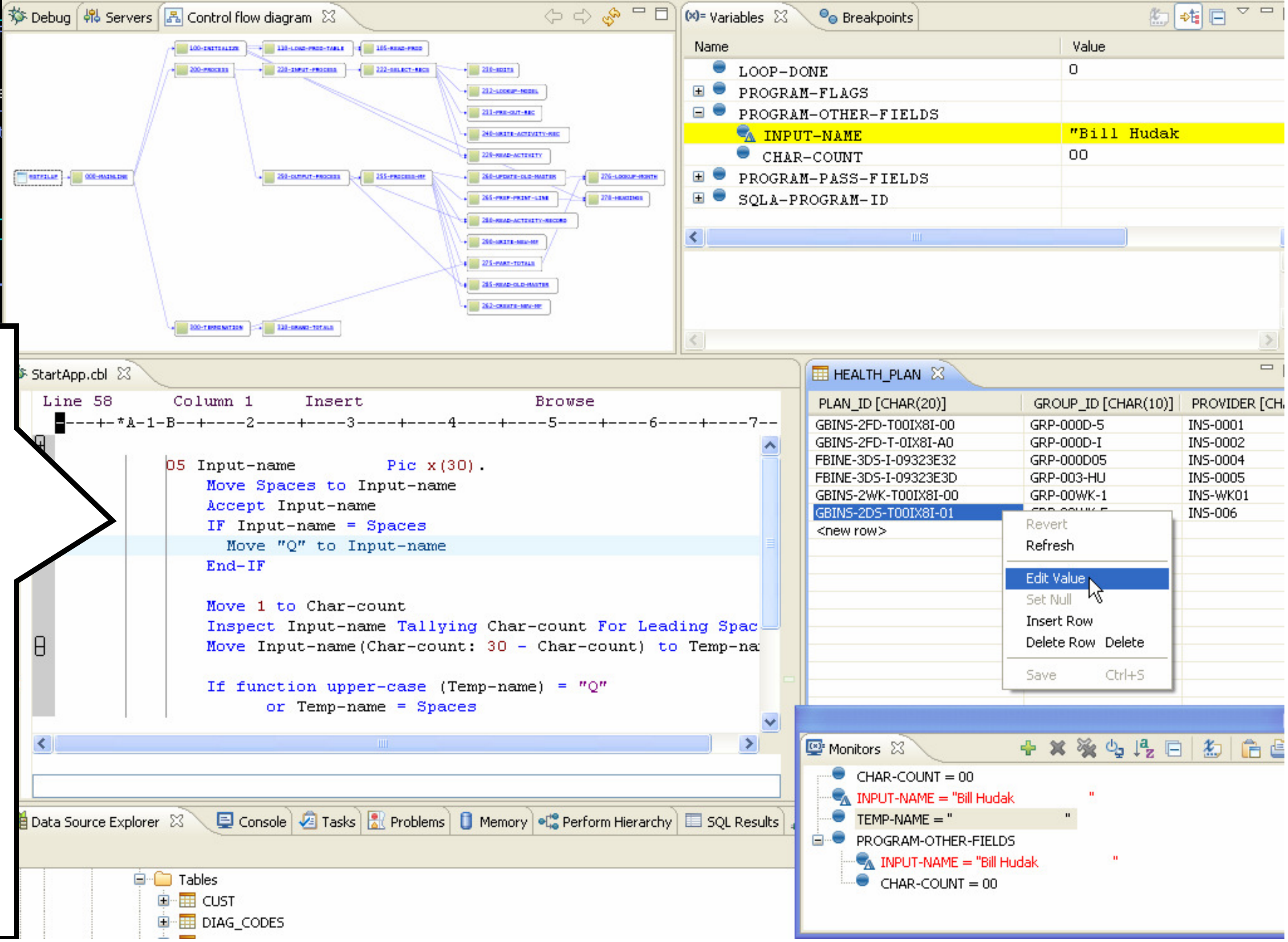
4. Responsive Desktop/Windows Environment

Same basic functionality since the ← late 1970's





```
  Menu  Functions  Confirm  Utilities  Help
 DSLIST        DDS0001.TEST.JCL                     Row 00001 of 00008
            Name     Prompt        Size   Created        Changed           ID
 _________ COMPLINK                  35  2009/05/07  2009/05/09 10:32:00  DDS0001
 _________ DATAIN                     5  2009/04/23  2009/04/23 10:27:26  DDS0001
 _________ DATAIN2                   12  2009/04/23  2009/04/23 15:15:33  DDS0001
 e________ HOSPCALC                  35  2009/05/09  2009/05/09 09:10:16  DDS0001
 _________ HOSPIN                    14  2009/05/07  2009/05/07 13:31:17  DDS0001
 _________ MORTGAGE                  25  2008/02/28  2008/04/03 10:52:04  DDS0069
 Command ===> ____________________
  F1=Help     F2=Split     F3=Exit     F
 F10=Left    F11=Right   F12=Cancel
 .  .  .  .  .  .  .  .  .  .  .  .  .
  Menu  RefList  RefMode  Utilities
                                          Edit
 ISPF Library:
    Project . . . DDS0001
    Group . . . . TEST      . . . .  ___
 Command ===> ____________________
  F1=Help      F2=Split        F3=Exit
 F10=Actions  F12=Cancel
 MA█      a
```



Name	Value
LOOP-DONE	0
PROGRAM-FLAGS	
PROGRAM-OTHER-FIELDS	
INPUT-NAME	"Bill Hudak
CHAR-COUNT	00
PROGRAM-PASS-FIELDS	
SQLA-PROGRAM-ID	

```
 Line 58       Column 1       Insert                    Browse
 ----+-*A-1-B--+----2----+----3----+----4----+----5----+----6----+----7--

       05 Input-name          Pic x(30).
          Move Spaces to Input-name
          Accept Input-name
          IF Input-name = Spaces
            Move "Q" to Input-name
          End-IF

          Move 1 to Char-count
          Inspect Input-name Tallying Char-count For Leading Spac
          Move Input-name(Char-count: 30 - Char-count) to Temp-na

          If function upper-case (Temp-name) = "Q"
                or Temp-name = Spaces
```

PLAN_ID [CHAR(20)]	GROUP_ID [CHAR(10)]	PROVIDER [CH
GBINS-2FD-T00IX8I-00	GRP-000D-5	INS-0001
GBINS-2FD-T-0IX8I-A0	GRP-000D-I	INS-0002
FBINE-3DS-I-09323E32	GRP-000D05	INS-0004
FBINE-3DS-I-09323E3D	GRP-003-HU	INS-0005
GBINS-2WK-T00IX8I-00	GRP-00WK-1	INS-WK01
GBINS-2DS-T00IX8I-01	[illegible]	INS-006
<new row>		

```
CHAR-COUNT = 00
INPUT-NAME = "Bill Hudak                "
TEMP-NAME = "                           "
PROGRAM-OTHER-FIELDS
   INPUT-NAME = "Bill Hudak                "
   CHAR-COUNT = 00
```

An RDz session with concurrent access to:

- Program analysis views
- Debugging functionality
- Edit and Browse of:
    - DB2 tables
    - IMS Database
    - VSAM files
    - QSAM files

Mitigating Factors

The following must be noted about this study:

1. **No use of custom ISPF Edit-Macros, etc.**
   - Many shops (and individual programmers within shops) have developed and use custom editing macros during their work.
     - These macros would in all likelihood improve the ISPF results.
     - To what degree is unknown…but possibly as much as: 5-10%

2. **No use of custom RDz Macros, PF-Keys or RDz Snippets**
   - These would in all likelihood improve the RDz results as much as: 3 – 5%

3. **Years of ISPF experience**
   - The ISPF development experience (10 years) of the participants is considerably more than their RDz experience
   - However, there are many shops with a mature developer-base that has an AVERAGE of 20+ years of ISPF experience
   - This discrepancy was mitigated as far as possible through the use of the detailed ISPF script (down to the PF-Key to be pressed)
   - But it is possible that another 10 years of ISPF experience would net an improvement in the ISPF results

Learn more at:

- IBM Rational software
- Rational launch announcements
- Rational Software Delivery Platform
- Accelerate change & delivery
- Deliver enduring quality
- Enable enterprise modernization
- Ensure Web security & compliance
- Improve project success
- Manage architecture
- Manage evolving requirements
- Small & midsized business
- Targeted solutions
- Rational trial downloads
- developerWorks Rational
- Leading Innovation
- IBM Rational TV
- IBM Business Partners
- IBM Rational Case Studies