



Increasing Mainframe Software Maintenance
Efficiency and Effectiveness: An Empirical
Study of Microcomputer versus
Mainframe-based Tools

White Paper

Abstract

Software metrics and measurement methodologies are commonly used in the development and tracking of projects. Although maintenance and debugging operations are known to comprise the major portion of the System Development Life Cycle, little work has been done to establish the costs involved in the execution of key tasks in these operations. In this study we investigate the influence of the type of maintenance tool on the efficiency and effectiveness of completing a set of commonly performed mainframe software maintenance tasks.

We explore the time necessary to accomplish certain mainframe software maintenance tasks using mainframe-based tools (TSO/ISPF on an IBM S/390 platform) versus micro-computer based tools (Micro Focus's REVOLVE/MFE on a Pentium based platform.) We envision the maintenance process as comprising a sequence of phases based on cognition in addition to simple task performance. These are:

- Mapping of business rules
- Understanding of code in this context
- Code manipulation

Steps 1 and 2, to be most efficiently executed, require more than a simple text editor. The cognitive processes in play involve code organization, representation of relationships and dependencies, search and display of needed information and code.

Our results indicate that for certain simple tasks (specifically, editing tasks) there is little difference in performance time between the two environments. However, for a number of tasks of a more complex nature, the microcomputer-based tools provided significantly increased efficiency. These findings were true across all levels of expertise tested.

The establishment of baselines for these tasks will allow for estimates of the cost to perform these tasks and a comparison of costs across the development environments examined (mainframe and PC.) Estimates could be based on the hourly costs for programmer/analyst time.

Contents

Introduction	2
Research Design & Methodology	3-8
Empirical Results	9-15
Statistical Analysis	16-21
Concluding Remarks	22-26
Citations	27
List of Appendices	28-57
Appendix A	
Appendix B	
Appendix C	
Appendix D	
Appendix E	
Appendix F	
Appendix G	

Introduction

Costs in software development and maintenance have been studied for decades. Numerous methods have been used in software application development. As these applications become larger, more complex, more expensive to create and maintain, and of greater strategic importance, it has become vital for an enterprise to have some method of measuring the costs involved during all phases of the software application's life cycle. The concept of measuring and controlling the Total Cost of Ownership (TCO) has become important. While it is now important to users to allocate Information technology expenditures wisely, it has been felt that measuring these costs is difficult [Jacobs, 1998.] The measurement of TCO is vital to an enterprise in that it allows for control of two of the greatest cost factors in IT: project risk management and maintenance costs [Aberdeen Group, 1998.]. This paper will concentrate on the measurement of certain of the costs involved in application maintenance, specifically the programmer time spent on debugging and maintaining code.

What is Software Maintenance?

Software maintenance is the performance of those activities required to keep a software system operational and responsive after it is accepted and placed into production. It is the set of activities which results in changes to the originally accepted (baseline) product set. These changes consist of modification created by correcting, inserting, deleting, extending, and enhancing the baseline system. The three types of software maintenance are usually classified as perfective, adaptive, and corrective [NBS, 1986.]

Perfective maintenance includes all changes, insertions, deletions, modifications, extensions, and enhancements that are made to a system to meet the evolving and/or expanding needs of the user. They are generally performed as a result of new or changing requirements, or in an attempt to augment or fine-tune the software.

Adaptive maintenance consists of any effort that is initiated as a result of changes in the environment in which a software system must operate. It accounts for about 20% of all the software maintenance efforts.

Corrective maintenance refers to changes necessitated by actual errors in a system. It accounts for 20% of all the software maintenance efforts and consists of activities normally considered to be error correction required to keep the system operational. We concentrate on perfective and corrective maintenance task measurement in this study. Software maintenance is important because studies have shown that over a standard application life cycle (averaging 6 years), the cost of maintenance is about 10 times the cost of the original development and testing. This amounts to a total of 50% - 80% of the total project cost being spent on maintenance [Brady, 1998.]

Research Design and Methodology

Research Setting

The programs used for maintenance purposes were derived from a special teaching system developed by a faculty member of DePaul University. This application was a simulation of a standard industry payroll system. In addition, actual programs, obtained from large organizations, were also used. Code consisted of COBOL, JCL, data, copybooks, and related information. The subjects did not see this code until test time.

Two sets of maintenance tasks were developed which were similar in terms of the maintenance function tested but different in code and description so as to minimize learning effects. These were common tasks spanning a range of general maintenance concepts. Subjects used one set of tasks when using mainframe tools. The same subjects used the corresponding set of tasks when using microcomputer tools. An attempt was made to create tasks that were as similar as possible for use in both environments.

Equipment/Environments

The testing environments were similar to those used in industry. Subjects worked at individual workstations situated on long counters or desktops.

Mainframe tools were used on a remote S/390 mainframe accessed using a TN3270 emulator and a dial up connection through an exchange server. Mainframe tools consisted of a standard COBOL compiler/editor environment (TSO/ISPF).

Microcomputer tools were used on personal computers. Due to problems in scheduling workstation time two sets of microcomputer subjects were run. The first set (2 subjects) worked on PCs with 64MB RAM and 166 MHz Pentium processors running under the Windows 95 operating system. The remaining 6 subjects worked with 64MB RAM and 350 MHz Pentium processors running under the Windows 98 operating system. PC software tools consisted of Micro Focus Mainframe Express and Revolve.

All subjects were exposed to the same mainframe environment and connections. Although the microcomputer tasks involved the use of two separate types of machines, the processors should not have differed tremendously in response speed. Subjects using the slower machines were Micro Focus experts who were expected to be more adept at use of the microcomputer tools, and thus their times should have been even quicker than they were, assuming that use of a faster processor would have made a difference.

Study Assumptions

The study approach was based on a situation normally found in standard IT environments:

Programmers do maintenance with whatever tools the installation management provides

Management obtains new tools that it believes will increase productivity

Programmers are trained in the new tools

The new tools are used

Management wishes to measure the increase in productivity that should justify the cost of obtaining and utilizing the product

Application Programs

There were actually two applications used in this research, with a combined total of over 150,000 lines of source code. The applications do traditional batch (JCL-driven) job processing, Read/Write/Rewrite of QSAM and VSAM files, reporting, file utilities (sort/merge, delete/define, etc.) and other standard MVS data-center batch requirements.

Companies from the banking, manufacturing, insurance and retail industries also donated application code used in this research. A complete working Payroll System developed by an instructor in the DePaul University, School of Computer Science, Telecommunications and Information Systems was, also used in its' entirety. These applications provided the subjects with a substantial code base that would mirror an actual business installation in functionality and complexity, yet be amenable to use in a short-term study.

Software complexity is an issue of importance in the measurement of the efficiency of maintenance and debugging tools. Halstead's metrics [Halstead 1977] are computed from properties of the source code.

Table 1 shows the results of calculating Halstead's Effort Metric for the application software used. This measure is available for use by anyone wishing to compare the test application against his or her own installed software base. As a metric it is considered to be one of the standards and more representative an indicator than the number of lines of source code [Fairley, 1990.] The Program Effort measure is an indicator of the number of mental discriminations required to implement the program, or put more simply - to read and understand it. This metric has been found to have a high degree of correlation with **effort** to debug and modify programs. This measure may be useful for anyone wanting to extrapolate these findings to their own application environment. It should be possible to measure the Effort metric to see how another application stands in relation to the code studied.

BigN_one	1184	N1 = total number of operators in a program
BigN_two	1158	N2 = total number of operands in a program
n_one	30	n1 = number of unique operators in a program
n_two	556	n2 = number of unique operands in a program
	5217.338	Program Length: $N = n1 \log_2 n1 + n2 \log_2 n2$
	21534.121	Program Volume: $V = (N1 + N2) \log_2 (n1 + n2)$
	0.0320092	Language Level: $L = (2 * n2) / (n1 * N2)$ (level of language abstraction)
	672747.62	Program Effort: $V/L = E = (n1 * N2 * (N1 + N2) * \log_2(n1 + n2)) / (2 * n2)$

Table 1. Halstead's Metrics for the application source code

Subjects

The test subjects included in the results each had between 2 and 25 years of TSO/ISPF mainframe COBOL development and maintenance background on large-scale MVS business software. This includes expert-level knowledge and experience in the following areas:

- ISPF navigation
- ISPF editing
- ISPF file allocation
- Job Control Language
- Job submission and review
- Mainframe ABEND research
- Software defect analysis

In addition, the participants were reasonably proficient in the following basic PC functions (as facilitated by the use of windows based PC tools and environments):

- Graphical editing
- Windows manipulation
- Mouse navigation
- Windows 95 usage
- Directory creation
- File copying
- Invoking (launching programs) from the desktop

A total of 8 subjects were involved in the test. Appendix D provides descriptive demographics of the subjects. Participants in the study were all experienced professionals and were chosen from the following groups:

Group 1. Micro Focus experts, experienced with both mainframe and microcomputer tools.

These subjects had extensive experience in the use of TSO and the Micro Focus products and PC environment. These subjects were included to examine the possible statistical significance of performance differences based on the level of expertise of PC tools users.

Group 2. Practicing senior maintenance programmers / COBOL instructors primarily familiar with mainframe tools – Micro Focus beginners

The remaining subjects were people with extensive background in COBOL programming. These people were all programmers, although some split their time between consulting and teaching. Their experience on mainframe environments varied from moderate to extensive. None of these subjects had used the Micro Focus software tools involved in the test, prior to the training.

These subjects learned to use these PC-based tools (Micro Focus MFE and Revolve) in a *compressed* time frame (typically 6 days of training, compressed down to 1.2). They also had completed up to 16 hours of additional interactive study using off-the-shelf training materials from Micro Focus. Thus, they were trained in the use of the PC based tools and were expert in the use of mainframe TSO environment and COBOL programming.

Methodology

This study fits the definition of a Quasi-Experimental study as outlined in most statistics texts, and specifically in the SPSS manual [Green et al, 1997.] The study borrows some elements of an ethnographic study in that the subjects were working in an environment that was as close as possible to their common work settings. Here we predict that skilled COBOL mainframe programmers, given a sufficient amount of training in the use of PC based tools, will take less time to analyze, debug, and correct coding problems when working in a PC based environment than will be needed when using a mainframe environment and tools (TSO/ISPF based).

The specific research question may be phrased so as to ask about the relationship between two variables. One variable is measured: elapsed time to complete task. The other is a group membership: task in TSO or PC environment. Following Kerlinger's methodology [1973] the basic principle of measuring a variable can include using the same subjects in both experimental groups. It should be pointed out that a random sample of subjects to fit both the demographics *and* the experimental design requirements (substantial time to be spent on training and testing – up to 32 hours) made it extremely hard to find subjects. Eight Subjects, all skilled in the use of TSO and COBOL programming, agreed to participate in the study. These subjects were given a series of programming tasks to be completed in the TSO/ISPF environment. Measurements were in minutes of elapsed time. All 8 subjects

were retested in the PC environment on a corresponding set of tasks devised to be similar to those used in the first set. These subjects were divided into two groups as follows:

- **Experts.** An "Expert" is defined as a person with at least 1 year's experience using graphical tools (mouse editing, windows navigation, etc.), who has gone through the required minimum training, and spent at least one month actively using the products on a daily basis
- **Beginners.** A "Beginner" is defined as a person with minimum graphical skills (not fluent with Windows manipulation, uncomfortable with mouse editing), who has taken some (minimum) training and has used the PC products for less than one month

The test variable was the ***elapsed time to complete task*** measured in minutes.

The study methodology included the following steps:

- A. Develop a set of common maintenance tasks that address areas to be measured and to be performed by the participants of the study
- B. Obtain mainframe programs to be used for maintenance tasks
- C. Provide Micro Focus training on the REVOLVE and MFE products used in the study to the subjects from Group 2 (described above.) An accredited Micro Focus trainer, using selected portions of the standard off-the-shelf Micro Focus, Inc. courseware adapted to the Benchmark provided product Training for this research. The training was a condensed version of a standard 6-day training course. Subjects were given approximately 8-10 hours of classroom training and additional self paced tutorials that took from 5 to 10 hours of additional time. These subjects spent an approximate average total of 15 hours on classroom and self paced training. TSO/ISPF training was not considered necessary given the participants' backgrounds
- D. Administer task set one (mainframe tools) Appendix A
- E. Collect and review results (time spent by task)
- F. Administer task set two (microcomputer tools) Appendix B
- G. Collect and review results (time spent by task)
- H. Analyze using descriptive statistics
- I. Present results

Subjects were given a workbook of tasks for each set. The complete description of tasks is included in Appendices A and B. These tasks were created to measure performance in certain categories. A list and description of the categories is included in Appendix C. Tasks were arranged one (or one group of similar items) to a page. Space was provided for answers. Subjects were monitored by the experimenters and starting and ending times were recorded. Subjects worked alone at their best individual speed. Any questions were directed to the monitors. Subjects could use any of the available on-line tools and had blank paper for note taking. Complete paper printouts of all source code were also

available to the subjects during the mainframe portion of the test. The two sets of tasks were devised to be as similar as possible. It was not felt that there would be any significant effect by testing subjects on the Mainframe environment first. Subjects were experts in this environment, knowing "how to" complete tasks. In the PC environment they had to determine how to do things, being less familiar with its features. Future studies will mix the order of presentations.

Empirical Results

The tables in Appendices E and F show the elapsed times taken by the subjects to complete each given task in both portions of the test. Missing values for subjects indicate that the subject did not complete the task. In most cases they simply ran out of time. A total of twelve hours of testing was expended on both sets of tasks and the subjects were limited in their availability. It should also be noted that the total time spent in the test environments was higher still than the elapsed times would indicate. The Mainframe test was slowed down by about 2.5 hours when the mainframe went down on 2 occasions. Missing data entries were in most cases explained on the workbook sheets. In a few cases the subject indicated that the task could not be performed or completed and indicated reasons. One of the subjects (Subject 6) did not complete the second half of the experiment (using PC based tools) due to illness. All missing or invalid time values were excluded from the final statistical analysis.

Task Pair 19 was excluded from the analysis. This task involved finding picture and value clauses for data items. Comments made during the test, analysis of resulting data, and discussion with some subjects led us to the conclusion that the instructions were not clear enough. Subjects only found first level values but did not search deeper into the code. This task should have taken about as long as Task 18 to complete. In addition, the task procedure appears to differ between the environments. This discussion points out the difficulties in designing such an experiment, which is to test subjects in a setting that is as close as possible to the real working world. All charts and tables show no values for this item.

All Charts bear item labels on the X-axes. Only every other label name is shown due to the limited amount of space available at the chart's bottom. Charts have blank entries for Task Item 19, which was excluded from analysis.

Elapsed Times - TSO Tasks

Chart 1. Elapsed Time for TSO Tasks (All Subjects)

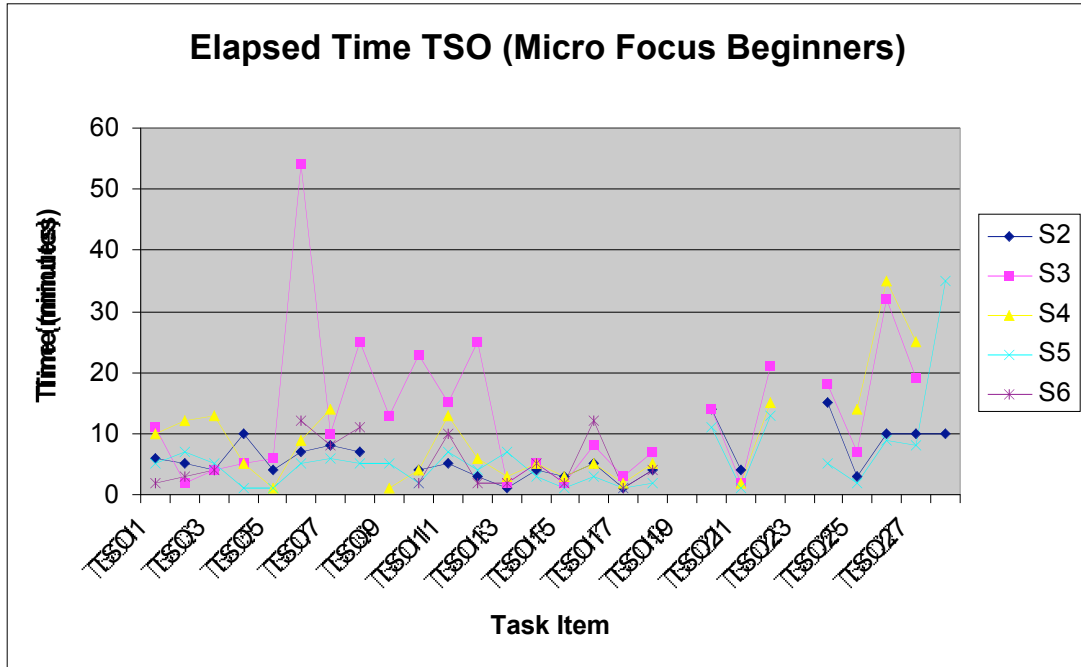


Chart 2. Elapsed Time for TSO Tasks (Micro Focus Beginner Subjects)

Elapsed times for each TSO task were fairly consistent across subjects as shown in Chart 1 and Chart 2. The times for most of the subjects were extremely close. Only Subject 3 appeared to vary greatly in some instances. This subject was wildly off (task item 6) due to misinterpretation of instructions. Statistical significance of the task was the same whether this datum was included in the analysis or not.

Elapsed Times - Revolve/MFE Tasks

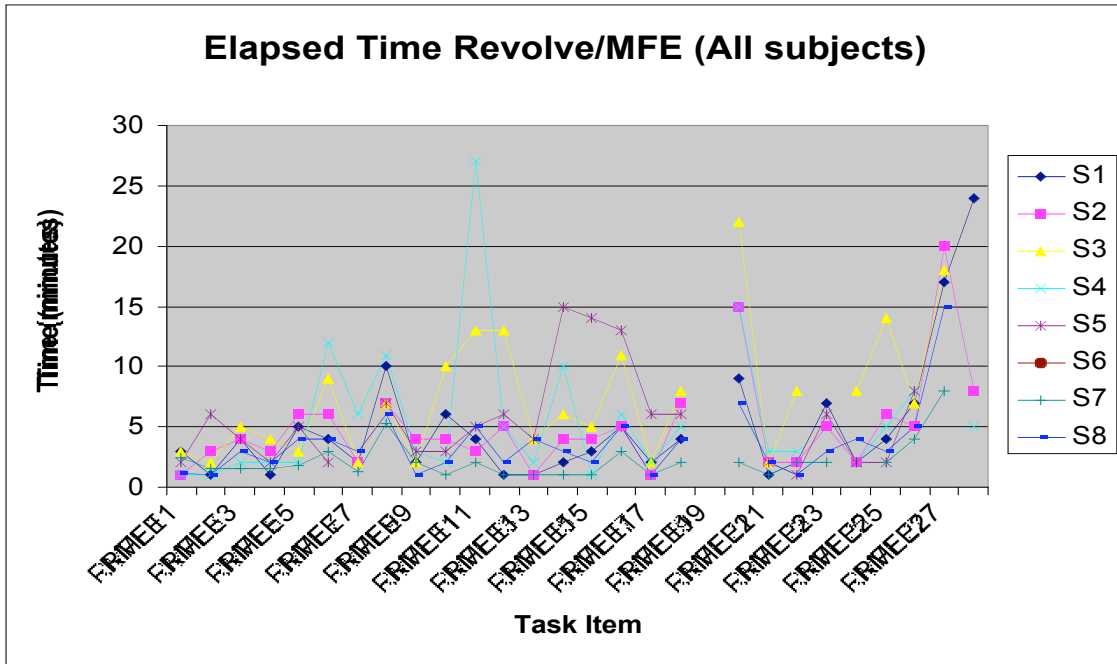


Chart 3. Elapsed Time for Revolve/MFE Tasks (All Subjects)

Though not as tight as the elapsed times for TSO tasks, the Revolve/MFE tasks were still fairly consistent across subjects (Chart 3.) Here it appeared that Subjects 3 and 4 were most outside the ranges of elapsed time per task on a few items. The last 5 items showed a great variation in recorded elapsed times. These items were the project tasks. Some subjects were not able to complete these in the time available to them.

Observation of subject performance indicated that a great deal of time was spent in analysis. It did not appear that the difference in PC Central Processor Unit speeds created any significant time delays in performance. (This is based on observation during the tests and in a review of data.) [This should be tested using timing/recording functions in future studies or controlled for in test setup. Subjects 7 and 8 used the slower 166 MHz PCs on this portion of the study and were quite fast on most tasks. Their task solution times would not likely have been significantly different had they used faster processors.]

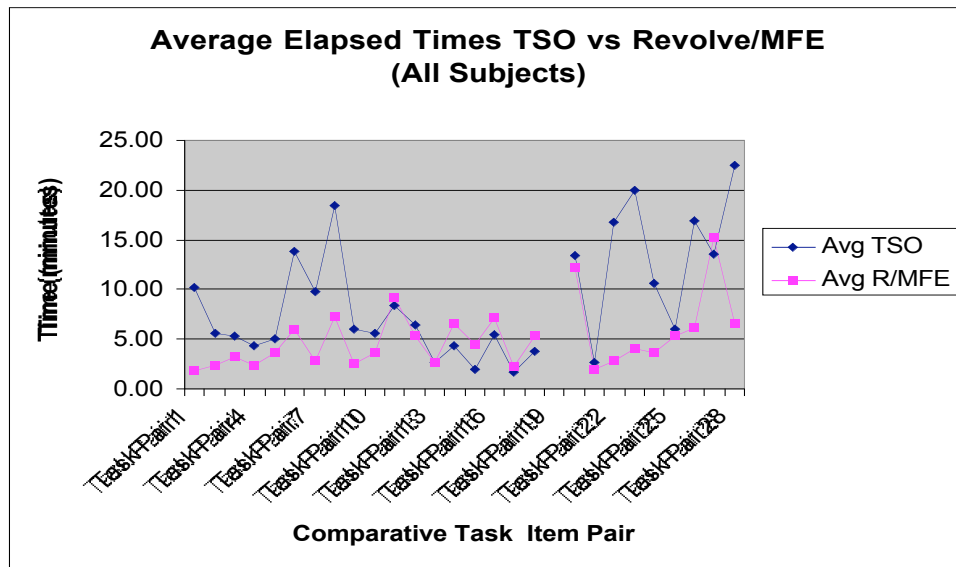
Comparison of Elapsed Times - TSO vs. Revolve/MFE Tasks

	Avg TSO	Avg R/MFE		Avg TSO	Avg R/MFE
Task Pair1	10.00	1.95	Task Pair15	2.00	4.29
Task Pair2	5.38	2.21	Task Pair16	5.50	6.86
Task Pair3	5.38	3.37	Task Pair17	1.63	2.14
Task Pair4	5.29	2.21	Task Pair18	3.88	5.14
Task Pair5	4.57	3.82	Task Pair19		
Task Pair6	13.75	5.71	Task Pair20	14.50	11.67
Task Pair7	9.50	2.71	Task Pair21	3.29	1.86
Task Pair8	20.86	7.61	Task Pair22	16.17	2.71
Task Pair9	5.50	2.43	Task Pair23	20.00	4.60
Task Pair10	5.50	4.00	Task Pair24	12.33	3.33
Task Pair11	7.88	8.43	Task Pair25	5.86	5.14
Task Pair12	6.38	4.71	Task Pair26	16.43	6.29
Task Pair13	2.50	2.43	Task Pair27	13.17	15.60
Task Pair14	4.00	5.86	Task Pair28	22.67	12.33

Table 2. Average Elapsed Times for TSO/Revolve Task Pairs (All Subjects)

Table 2 contains the average elapsed times per task for all subjects counted. TSO and Revolve/MFE tasks were matched and compared. Comparison on a task by task basis is presented in Chart 4.

Chart 4. Average Elapsed Times TSO v Revolve/MFE (All Subjects)



Here we see that task items 1 through 9 took longer on average (in some cases considerably longer) to complete when working in the TSO environment. These items were the research and analysis tasks. These tasks required some knowledge of relationships among program components.

Task items 10 through 18 took roughly the same amount of time for the non-experts, no matter the environment. These were editing tasks. It appears that once situated in a file the editing operations take roughly the same amount of time in either environment.

Task items 20 through 26 and item 28 also took longer on average (in some cases considerably longer) to complete when working in the TSO environment.

- Items 20 and 21 were edit tasks involving a long sequence of replacements and copies between files, respectively. This involved working in a multi-file environment.
- Items 22 and 23 involved compiling and finding COBOL syntax errors. These took considerably less time to complete in the Revolve/MFE environment.
- Item 24 involved compiling a portion of an application that was affected by a maintenance modification. This item was done on average in considerably less time using Revolve/MFE environment.
- Item 25, batch integration testing, was done on average in slightly less time in the Revolve/MFE environment.
- Item 26, testing COBOL logic as a result of variable data debugging, was done in considerably less time in the Revolve/MFE environment.
- Item 27, testing COBOL logic with intra-module execution path verification, took slightly longer to do on average in the Revolve/MFE environment. This involved the checking of logic paths in CNTRLBK.
- Item 28 was an extended maintenance project using the payroll register application. Here again the average times are considerably shorter when working in the Revolve/MFE environment.

A comparison of average elapsed times for the completion of these tasks by Micro Focus Beginner Subjects is shown in Chart 5. Here it can be seen that the results found are similar to the trends found for all subjects. Those subjects who were not Micro Focus Experts were introduced to use of the software in this study. Their level of expertise is at an entry level yet the results found are similar to those that are found across all subjects.

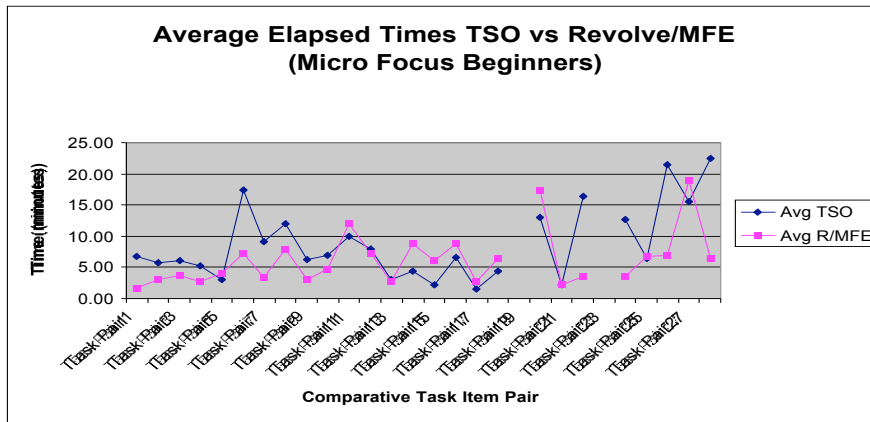


Chart 5. Average Elapsed Times TSO v Revolve/MFE Micro Focus Beginner Subjects)

Here we see that task items 1 through 4, and items 9 and 10 took longer on average to complete when working in the TSO environment. Item 5, which was a search for modules in a calling chain that invoked a certain subroutine, took slightly less time to complete when working in the TSO environment. Here familiarity with TSO may have trumped the PC based tools due to shortened training and experience. Items 6, 7, and 8 took considerably longer on average to complete in the TSO environment. These items were the research and analysis tasks. These tasks required some knowledge of relationships among program components which PC based tools may have helped to highlight.

Task items 11 through 13 and 17 took roughly the same amount of time for the Beginners, no matter the environment. Items 14 through 16 and 18 took a bit longer to do in the PC based environment. These were simple editing tasks. It appears that once situated in a file the simplest editing operations take roughly the same amount of time in either environment

Task items 21 through 26 and item 28 also took longer on average (in some cases considerably longer) to complete when working in the TSO environment.

- Items 20 and 21 were edit tasks involving a long sequence of replacements and copies between files, respectively. This involved working in a multi-file environment. Task item 20 took slightly longer on average to complete in the PC based environment, while item 21 took about the same amount of time on average to complete regardless of the environment.
- Items 22 and 23 involved compiling and finding COBOL syntax errors. Task 22 took considerably less time to complete in the Revolve/MFE environment. Task 23 was inconclusive due to a lack of data.
- Item 24 involved compiling a portion of an application that was affected by a maintenance modification. This item was done on average in considerably less time using Revolve/MFE environment.
- Item 25, batch integration testing, was done in about the same amount of time no matter the environment.

- Item 26, testing COBOL logic as a result of variable data debugging, was done in considerably less time in the Revolve/MFE environment.
- Item 27, testing COBOL logic with intra-module execution path verification, took slightly longer to do on average in the Revolve/MFE environment. This involved the checking of logic paths in CNTRLBK.
- Item 28 was an extended maintenance project using the payroll register application. Here again the average times are considerably shorter when working in the Revolve/MFE environment.

When making a final comparison, we see that resultant trends were similar whether the subjects were experienced or had recently started using the PC based tools.

Statistical Analysis

Descriptive Statistics

Descriptive statistics were computed for each pair of tasks studied. These are the corresponding pairs of tasks performed under the mainframe (TSO/ISPF) and PC (Mainframe Express/Revolve) environments. The descriptive statistics on these tasks indicate the numerical representation of the shape of the distribution.

There were various problems with the data. Missing values represent a number of different situations. A zero could not be used in these cases because of the implications it would have for statistics on elapsed times. Some subjects could not complete all tasks due to fatigue. The test sessions took as long as twelve hours.

A paired sample t Test was first run. The Mann Whitney U Test was also performed. Results of both show significant findings for the same tasks (or task pairs tested.) These tasks were cognitive analytical tasks, rather than simple editing functions, and generally took more time to do than the simple editing tasks. Missing data values were eliminated from statistical analysis. Only entries where data values were present were used. This is a feature of the Mann Whitney test.

t-Test -- the standard t-test compares the means for two groups of test cases.

The paired samples t-test compares the means of two variables that represent the same group at different times. It computes the differences between values of two variables for each case and tests whether these averages differ from 0. If the correlation is low and the significance value is high, the independent samples t-test may be used. In a paired samples, test means and 95% confidence intervals are examined. If the confidence interval for the mean difference does not contain zero, this indicates that the difference is significant. A low value for 2-tailed significance also indicates a significant difference between the variables. Statistics calculated for each variable include:

- mean
- sample size
- standard deviation
- standard error for the mean

Statistics calculated for each pair of variables include:

- correlation
- average difference in means
- t-test
- standard deviation and standard error of the mean difference
- confidence interval for the mean difference

This test is commonly used in medical studies to test the effect of some treatment on a group of patients. Patients are measured at the beginning of the study, treated, and then measured again. The general scheme is as follows:

- Initial Measurement
- Intervention
- Post Measurement

Thus, each subject is a member of two groups.

In this study, we measure all subjects for completion of a number of tasks initially, train them in the use of a new tool set, and then measure them again for completion of each of a new set of tasks. Here the intervention consists of not just the training on the new tool, but its use in the new PC environment.

Table 3 and Table 4 show the task pairs found to have significant mean differences at the .05 level when tested across all subjects and Micro Focus Beginner Subjects respectively. The same pairs listed in Table 4 (Micro Focus Beginner Subjects) are also listed in Table 3, which has 3 additional significant pair entries.

		Paired Differences					Sig
		Mean	Std. Deviation	Std. Error Mean	t	df	(2 - tailed)
Pair 1	TSO1 - MFE1	9.1929	6.2331	2.3559	3.9020	6	.008
Pair 2	TSO2 - MFE2	3.5000	3.5940	1.3584	2.5770	6	.042
Pair 3	TSO7 - MFE7	7.6250	1.3205	.5391	14.1440	5	.000
Pair 4	TSO8 - MFE8	15.4583	13.1210	5.3566	2.8860	5	.034
Pair 5	TSO22 - MFE22	13.3333	2.1602	.8819	15.1190	5	.000
Pair 6	TSO23 - MFE23	17.5000	.7071	.5000	35.0000	1	.018
Pair 7	TSO24 - MFE24	9.8000	6.6106	2.9563	3.3150	4	.030

• **Table 3.** Paired Samples t Test Descriptive Statistics (All Subjects) Significant results

•

		Paired Differences					Sig
		Mean	Std. Deviation	Std. Error Mean	t	df	(2 - tailed)
Pair 1	TSO1 - MFE1	6.2000	2.3875	1.0677	5.8070	4	0.004
Pair 2	TSO7 - MFE7	7.0000	1.1547	.5774	12.1240	3	0.001
Pair 3	TSO22 - MFE22	12.0000	.8165	.4082	29.3940	3	0.000
Pair 4	TSO24 - MFE24	11.2500	6.6521	3.3260	3.3820	3	0.043

Table 4. Paired Samples t Test Descriptive Statistics (Micro Focus Beginner Subjects Only) Significant results

Results of the t Test indicate that Micro Focus Beginner Subjects (see Table 4) completed tasks using the PC based tools with mean faster elapsed times at a statistically significant level for the following task pairs:

- Pairs 1 and 2: TSO1/MFE1 and TSO7/MFE7 (research and analysis tasks)
- Pairs 3 and 4: TSO22/MFE22 and TSO 24/MFE24 (compile and test operations)

Results of the t Test indicate that for **All** subjects (see Table 3) the same statistically significant differences that were mentioned above hold true. In addition, the following items are found to be statistically significant:

- Pair 2: TSO2/MFE2 (research and analysis task)
- Pair 4: TSO8/MFE8 (research and analysis task)
- Pair 7: TSO23/MFE23 (compile and test operation.)

Mann-Whitney Test

The Mann-Whitney U Test is the most popular of the two-independent-samples tests. It tests that two sampled populations are equivalent in location when values are placed in rank-ordered lists. Observations from both groups are combined and ranked. Calculations are done on the number of times a score from each group precedes a score from the other group. The Mann-Whitney U statistic is the smaller of these 2 numbers.

The Mann-Whitney test measures elapsed times for each task under two groups. Here we analyze the results obtained by each Group (the grouping Variable.) Group 1 is the mainframe environment. Group 0 is the PC environment. The two groups are examined and data ranked. Of interest were both the overall response times for all subjects and, especially, the response times of the Beginner Subjects, who were trained in the use of Micro Focus PC based tools as a part of this study.

Mann Whitney U test scores on test variables are converted to *ranks* independent of membership in any groups. The mean ranks of the two groups are then compared to see if there are any significant differences. We are interested primarily in differences between mean ranks of the elapsed times between the TSO and PC environments.

Because ranked scores are used, the distributions do not have to be of any particular form. The descriptive statistics on these tasks indicate the numerical representation of the shape of the distribution. Distributions are generally asymmetrical (positively skewed.) This militates against applying an analysis that assumes a normal distribution. Use of the Mann Whitney test thus alleviates problems that would occur if we relied on statistical tests that assume a normal distribution.

We present both Mann Whitney U Test results indicating ranking relationships on test variables, as well as raw elapsed time scores and distributions. Without the latter, a reader would have little idea of what the raw results were like.

The following tables list rank data for each of the tasks found to have a statistically significant test result. (All tasks not statistically significant are deleted from this table. Appendix F contains the Mann-Whitney Rank table for **all** tasks tested.) The mean rank for each task lists the average of the ranks for each group. Similar values would indicate similarity in Group location. Small significance values ($< .05$) indicate that the two Groups have different locations when placed in rank-ordered lists. This would indicate a significant difference in elapsed time based on the tool/environment used.

		ASK1	ASK2	ASK7	ASK22	ASK24	ASK26
Mann - Whitney U		.500	.000	.500	.000	.500	.000
Wilcoxon		1.500	6.000	1.500	8.000	3.500	2.000
Z		2.843	2.335	3.068	3.025	2.531	2.646
Sig. (2-tailed)	Asymp.	.004	.020	.002	.002	.011	.008
	Exact Sig. [2*(1 - tailed Sig.)]	.002	.021	.001	.001	.009	.007

- a. Not corrected for ties
- b. Grouping Variable: GROUP

Table 5. Mann Whitney U Test (All Subjects) Significant results

		ASK 1	ASK 7	ASK 22	ASK 24	ASK 26
Mann - Whitney U		.500	.500	.000	.000	.000
Wilcoxon		7.500	0.500	5.000	6.000	5.000
Z		2.298	2.522	2.470	2.303	2.627
Sig. (2-tailed)	Asymp.	.022	.012	.014	.021	.009
	Exact Sig. [2*(1 - tailed Sig.)]	.017	.010	.016	.032	.008

- a. Not corrected for ties
- b. Grouping Variable: GROUP

Table 6. Mann Whitney U Test (Micro Focus Beginner Subjects) Significant results

Table 5 results indicate that under the Mann Whitney test, statistically significant differences in mean elapsed time exist when performing tasks in the TSO and PC environments. The resulting significant pairs are the same for All Subjects as they were under the t Test with the following exceptions:

- Task pairs 8 and 23 are not found to be significant
- Task 26 (mainframe compile and test operation) is found to have a statistically significant difference between the group means.

Table 6 results indicate that under the Mann Whitney test, statistically significant differences in elapsed mean times also exist for the Micro Focus Beginner Subjects. These are the same as those found under the t Test with the following exceptions:

- Task 26 (mainframe compile and test operation) is found to have a statistically significant difference between the group means.

The Mann Whitney U Test yields results similar to those of the paired sample t Test. For the reasons listed in the statistical analysis section (especially those having to do with sample size, incomplete data, and methodology) the Mann Whitney U Test will be used to make the conclusions in Section 5 following.

Table 7 shows the results of running the Mann Whitney U Test on the data, when the Experience of the Subjects with Micro Focus tools is examined. Here we look at the Subject Groups as being Expert and Beginner as described in previous sections of the report. The results indicate that differences in expertise are statistically significant (<.05) for the following Task Item Pairs.

Item Pair	Significance (2 tailed)
12	.020
14	.041
16	.029
18	.043
20	.019
21	.038

Table 7. Mann Whitney U Test Results Experience Effect

Concluding Remarks

1. The results of this study suggest that increased efficiency can be derived from using microcomputer-based software tools for mainframe software maintenance purposes.
2. In many cases observable trends in elapsed times to complete tasks indicate that the PC based tools allow for faster completion of tasks. In addition, statistical analysis indicates that at statistically significant levels when measuring the mean elapsed times for completing similar tasks under 2 environments (TSO/ISPF vs. PC based) the PC based tools allow for faster completion of a number of research, analysis, compile and test operations. In general, such results are statistically significant when the maintenance tasks are of an analytical nature. It appears that even simple editing tasks may be completed in shorter times using PC based tools when experience is factored in, but this would require further testing. The task pairs found to be different at significant levels are listed in the preceding section on statistical analysis.
3. For all tasks found to have a statistical difference in the mean completion time under Mann Whitney, the overall time savings can be projected as follows:
 - For All Subjects, on all significant tasks, the difference in mean time expended working under the PC based (MFE/Revolve) environment is only 45% of the time spent working under the TSO/ISPF environment on similar tasks. These figures should be replicable for any similar mix of subjects with varying degrees of expertise in the PC based tools, working on similar types of tasks.
 - For Micro Focus Beginner Subjects, the difference in mean time expended working under the PC based (MFE/Revolve) environment is only 50% of the time spent working under the TSO/ISPF environment on similar tasks. These figures should be replicable for any similar group of subjects with introductory training in the PC based tools, working on similar types of tasks.
4. A short amount of training and practice in the use of the PC based tools (MFE/Revolve) can return significant savings in elapsed time to complete certain research, analytic, and compile and test operations. Statistical outcomes indicate that there is a significant difference in task performance between Expert and Beginner subjects on a number of tasks.
5. There is some statistical evidence that experience in using the PC based tools yields increased timesaving over the savings garnered by new users of the tools. For Expert Users of the PC based tools the difference in mean time expended working under the PC based (MFE/Revolve) environment is only 45% of the time spent working under the TSO/ISPF environment on similar tasks.
6. There was no statistically significant difference in the mean times for completing simple editing tasks under TSO and PC based environments. These are relatively simple operations requiring location within a file. Differences between Expert and

Beginner Subjects show a slight advantage to use of the PC based tools. This should be tested more fully.

Productivity Gains

Analysis of the results indicates that there are average productivity gains which might accrue to use of the PC based Micro Focus tools. The specific gains obtained on statistically significant tasks have been stated above. The following tables (Table 8 through Table 11) show productivity gains indicated by the data. These gains are shown for all subjects and in some cases for Expert subjects when comparing total average task times in the TSO and RMFE environments. These will need further study to test for validity.

AVERAGE ELAPSED TIMES			
	TSO	RMFE	
	All Users	All Users	Experts
Task Pair1	10	1.95	2.22
Task Pair2	5.38	2.21	1.17
Task Pair3	5.38	3.37	2.86
Task Pair4	5.29	2.21	1.50
Task Pair5	4.57	3.82	3.58
Task Pair6	13.75	5.71	3.67
Task Pair7	9.5	2.71	2.08
Task Pair8	20.86	7.61	7.08
Task Pair9	5.5	2.43	1.67
Avg Total	80.23	32.02	25.82667
Productivity Gains		60%	68%

Table 8. Productivity Gains Obtained on Analysis Tasks

AVERAGE ELAPSED TIMES			
	TSO	RMFE	
	All Users	All Users	Experts
Task Pair10	5.5	4	3.00
Task Pair11	7.88	8.43	3.00
Task Pair12	6.38	4.71	1.33
Task Pair13	2.5	2.43	2.00
Task Pair14	4	5.86	2.00
Task Pair15	2	4.29	2.00
Task Pair16	5.5	6.86	0.00
Task Pair17	1.63	2.14	4.33
Task Pair18	3.88	5.14	3.33
Task Pair20	14.5	11.67	6.00
Task Pair21	3.29	1.86	1.33
Avg Total	57.06	57.39	28.33
Productivity Gains		0%	50%

• **Table 9.** Productivity Gains Obtained on Editing Tasks

AVERAGE ELAPSED TIMES

	All	Users
	TSO	RMFE
Task Pair22	3.29	2.71
Task Pair23	20.00	4.60
Task Pair24	12.33	3.33
Avg Total	35.62	10.64
Productivity Gains	71%	

Table 10. Productivity Gains Obtained on Compile Tasks

	AVERAGE ELAPSED TIME		
	TSO	RMFE	
	All Users	All Users	Experts
Task Pair25	5.86	5.14	3.00
Task Pair26	16.43	6.29	5.33
Task Pair27	13.17	15.60	13.33
Avg Total	35.46	27.03	21.67
Productivity Gains	24%		40%

Table 11. Productivity Gains Obtained on Testing Tasks

Future Work

The following are future projects that could continue this inquiry:

- Testing completely for equivalence between the 2 sets of tasks could refine the methodology and lead to elimination of those task pairs, such as Task 19, that are not found to be fully equivalent or have confusing instructions.
- Automated tools could be developed to record all timings and actions taken.
- The measures of cognitive complexity could be refined (including Halstead, McCabe other complexity measures) and comparison to other extensive application source code made.
- Expanded study on productivity gains over all task types is warranted.
- Methodology could move more in the direction of Ethnographic study, if suitable volunteers could be found. This would allow a study of the use of the software in a full production, real-world environment. In this way, vital components of the environment such as corporate culture, work relationships, integration of tools, and others could be observed and taken into account.

One of the major problems in conducting this study revolved around the need to find subjects who were experienced COBOL programmers and who could devote up to 32 hours to the entire process. This included time for compressed training , practice tutorials , and testing. Subjects had to work all of this in around their busy workweeks. There was no way to measure the effect of fatigue on subject performance in this study. This is a factor that should also be addressed.

- Subjects could also be given the full Micro Focus training on MFE/Revolve and testing could then be done against a broader group of subjects. The effect of expertise would be tested in a more controlled manner.
- Refine testing procedures to shorten the amount of time needed to complete tasks. This experiment is a stepping off point to refining procedures used in a difficult measurement task.
- Refine test to include a mixed order of problem presentation (Mainframe vs. PC based environments.) We suspect that there will be little or no significant difference as completion of a task in one environment involves little that can be carried over into the other environment.

Citations

Aberdeen Group. Risk Management and Total Cost of Ownership: Marketplace Realities. Executive White Paper, March, 1998.

Brady, James. "Automated Y2K Testing." *Enterprise Systems Journal*, v**13**(6), pp. 92-95, June, 1998.

Coffee, Peter. "Coders Squander Their Most Costly Asset: Time." *PC Week*, v**12**(26), p. 16, 7/3/95.

Fairley, Richard. Software Engineering Concepts. McGraw-Hill, N.Y, N.Y, 1990.

Green, Samuel B., Slackened, Neil J., and Key, Theresa M. Using SPSS for Windows. Prentice Hall, Inc. Upper Saddle Neck, NJ, 1977.

Halstead, M. *Elements of Software Science*, Elsevier, N.Y. N.Y., 1977.

Jacobs, A.. "Complex Costs Evade Control." *Computerworld*, v**32**(38), pp. 61-62, 9/21/98.

Kerlinger, F.N., Foundations of Behavioral Research, 2nd Edition, 1973.

NBS Special Publication 500-130, October 1986.

List of Appendices

- A. Task list one - Mainframe Tasks ISPF/TSO
- B. Task list two - Revolve/MFE
- C. Categorization of Mainframe and PC Tasks
- D. Demographics of participants
- E. Elapsed Time Results - TSO tasks
- F. Elapsed Time Results - Revolve/MFE tasks
- G. Paired Samples Test Result for All Subjects

Appendix A. Task List One - ISPF/TSO

The following is a list of all Mainframe TSO/ISPF tasks tested in the study. Each item is numbered sequentially. Some of these items are also marked by a Task Item number, in this case TSOx or TSOxx, following the sequential number. All tasks bearing a sequence number of this type are a part of a matched pair. The pair is identified by the xx following the TSO. The item is paired with a corresponding task from the MFE/Revolve test set in **Appendix B**.

II. Mainframe - Specific Research and Analysis tasks

TSO1 List the program(s) that READ the file TEST.PRODFILE.DATA:

TSO2 List the program(s) that update (WRITE and/or REWRITE) the file: TEST.EMPVSM.DATA:

TSO3 List the DATA DIVISION declarations (01-level group variable names) used to access the file:

TEST.EMPVSM.DATA:

TSO4 List the program(s) that utilize the COPY file: XPRD036F

TSO5 List the modules (program-names) in the Calling chain, that invoke the subroutine ACCT70PA –

invoked out of the ACCTSBAT job:

TSO6 In a reengineering project to provide online access to data, three batch COBOL programs

(BATCHRPT, DPCOBEXR and MSTFILUP) will be rewritten to become CICS online programs.

There are several COBOL verbs that are not allowed by CICS. List the program names and line #s

of the source that contain any following verbs: ACCEPT, SORT, DISPLAY:

TSO7 There was a calculation problem in the output from ACCT17PA last night. The errant arithmetic is in the paragraph P0620-ADD-CREDIT. List the execution path (paragraph) names

from the beginning of the program to the paragraph: P0620-ADD-CREDIT:

TSO8 The field W90-VENDOR-FILE-REC-ID – part of the group field:

W90-VENDOR-FILE-WORK-AREA.

03 W90-OPER-ID PIC X(3) ...

03 W90-VENDOR-FILE-REC-ID PIC X(3) ...

In ACCT18PA must be increased in size from 8 to 20 bytes. Name the application elements that are affected:

TSO9 What data (list the variable name through which the data is accessed) is contained in bytes: 4 - the file MFOCUS.PROD12C0:

III. Mainframe - Development Operations – ISPF Editing

TSO10 Loading a program into the editor: Edit the following programs. From within the PDS member list, in the ISPF Editor (Load each program into the ISPF editor, Insert one new blank line at the bottom of the file, and exit out of the ISPF editor) – save all changes

TSO11 Scrolling and Navigating within source files. From within, MSTFILA – Type a percent sign in Column 7 – for the following line/columns in the source:

- The 1st **SELECT** statement
- The first 05-field name in the **SORT-WORK-FILE FD**
- The line that contains the literal value **FEB**
- The **GOBACK** statement line, in **000-MAINLINE**
- The line at the top of file - **IDENTIFICATION DIVISION** statement.
- The bottom of file on the line that contains **AFTER ADVANCING 2.**
- Beginning of Working Storage Section: **WS-OLD-MF-REC.**
- Beginning of Linkage Section: **MSTF-PARMS.**
- **PROD-LINE-FILE-IN - FD**

- First procedural line, at the beginning of paragraph: **100-READ-PROD**

TSO12 Insert Operations - From within the COBOL program MSTFIL0 – insert a single blank line below (after) the following lines (save all changes):

- Bottom of file
- Top of file
- **FILE CONTROL**
- 002400 **FD OLD-CLAMS-MF-IN**

SELECT PROD-LINE-FILE-IN ASSIGN TO UT-S-PRODFILE.

- 030300 **100-INITIALIZE.**
- Beginning of **Working Storage Section**
- Beginning of **Linkage Section**
- Beginning of **Procedure Division**
- **110-LOAD-PROD-TABLE** paragraph

TSO13 Delete Operations:

Single-Line Delete Operations - From within the COBOL program MSTFIL2 – delete the following single lines (save all changes):

- **SELECT PROD-LINE-FILE-IN ASSIGN TO UT-S-PRODFILE.**
- **DATE-WRITTEN. JULY 1996.**
- **01 OLD-MF-RECORD PIC X(100).**
- **FD OLD-CLAMS-MF-IN**
- 01 WS-OLD-MF-REC.

TSO14 Multi-Line Delete Operations - From within the COBOL program MSTFIL3 - Delete the following contiguous source lines (save all changes):

- All **SELECT ASSIGN** statements
- The entire **REPORT-FILE-OUT FD**
- The entire **WS-PROD-LINE-REC. 01** group
- The entire **WS-DETAIL-LINE. 01** group
- All **FILLER** lines **under MONTH-DATA (under MONTH-TABLE-DEFINITION)**
- The entire **100-INITIALIZE** paragraph
- The **READ** statement in the **229-READ-ACTIVITY** paragraph
- The entire **275-CLAM-TOTALS** paragraph

- The **IF NOT EOF-SRT** statement in **280-RETURN-ACTIVITY-RECORD**
- The entire **290-WRITE-NEW-MF** paragraph.

Copy operations

TSO15 Single-Line Copy - From within the COBOL program MSTFIL4 – copy the following single lines to the top of file (save all changes):

- **SELECT PROD-LINE-FILE-IN ASSIGN TO UT-S-PRODFILE.**
- **DATE-WRITTEN. JULY 1996.**
- **01 OLD-MF-RECORD PIC X(100).**
- **FD OLD-CLAMS-MF-IN**
- **01 WS-OLD-MF-REC.**

TSO16 Multi-Line Copy - From within the COBOL program MSTFIL5 - Copy the following blocks of lines (contiguous source lines) **to the top of file**(save all changes):

- All **SELECT ASSIGN** statements
- The entire **REPORT-FILE-OUT FD**
- The entire **WS-PROD-LINE-REC. 01** group
- The entire **WS-DETAIL-LINE. 01** group
- All **FILLER** lines **under MONTH-DATA (under MONTH-TABLE-DEFINITION)**
- The entire **100-INITIALIZE** paragraph
- The **READ** statement in the **229-READ-ACTIVITY** paragraph
- The entire **275-CLAM-TOTALS** paragraph
- The **IF NOT EOF-SRT** statement in **280-RETURN-ACTIVITY-RECORD**
- The entire **290-WRITE-NEW-MF** paragraph.

TSO17 Replicate Operations - From within the COBOL program MSTFIL8 – Duplicate the following single lines (save all changes):

- **SELECT PROD-LINE-FILE-IN ASSIGN TO UT-S-PRODFILE.**
- **DATE-WRITTEN. JULY 1996.**
- **01 OLD-MF-RECORD PIC X(100).**
- **FD OLD-CLAMS-MF-IN**

- **WS-OLD-MF-REC.**

TSO18 Multi-Line Replicate Operations - From within the COBOL program MSTFIL9 - Duplicate the following contiguous source lines (save all changes):

- All **SELECT ASSIGN** statements
- The entire **REPORT-FILE-OUT FD**
- The entire **WS-PROD-LINE-REC. 01** group
- The entire **WS-DETAIL-LINE. 01** group
- All **FILLER** lines **under MONTH-DATA (under MONTH-TABLE-DEFINITION)**
- The entire **100-INITIALIZE** paragraph
- The **READ** statement in the **229-READ-ACTIVITY** paragraph
- The entire **275-CLAM-TOTALS** paragraph
- The **IF NOT EOF-SRT** statement in **280-RETURN-ACTIVITY-RECORD**
- The entire **290-WRITE-NEW-MF** paragraph.

TSO19 "FIND" Operation within the context of mixed application research and development activity.

From within the program - RCPOST:

- **Position your cursor on Paragraph: 301-MATCH-CUSTOMER and find the following:**

The value clause for the following 88-level data items:

- RETURN-CHECK _____
- INVOICE-REVERSAL _____

The picture clause for the following data items:

- WK06-TRAN-CODE _____
- WH01-DIV-TRADE-STYLE _____

TSO20 Discretionary and Global "Change" Operation - From within, RCPOSTB – Change the following (save all changes):

- Replace the 1st occurrence of WA01-CUSTOMER-BAL with WS01-WORK-FLD-NUM
- Replace all occurrences of NEW-RMST2-CUSTOMER-NO with NEW-RMS6TH-FLD-RED
- Replace all occurrences of 6000-CLNT-TOTALS with 5980-MAX-PERFORM-INTRST

- Replace the last occurrence of WS01-PREVIOUS-RECORD with WS88-REC-EXCPTN
- Replace the 3rd and 6th occurrences of LW40-ABEND-CODE with WKS-ABEND-RTN
- Replace all occurrences of NEW-RMST2-RECEIVABLES-BALANCE with WS-STOP-PYMNT
- Replace the 1st & last occurrence of LW20-INIT-DATE-PARMS with LW40-ABEND-CODE
- Replace every other occurrence of TRANS-FILE-IN with WS-DATA-RECORD-IN
- Replace all occurrences of NEW-CUSTOMER-MASTER-REC with OLD-CUST-MASTER-REC
- Replace the 1st occurrence of WA01-WORK-BALANCE with – WS-INTRNAL-BALANCE-99

TSO21 Copy multiple lines from one file to another

- From within **MSTFILUP** copy the Working-Storage 01-group field: **WS-OLD-MF-REC** (copy the entire 01-group) to the Working Storage Section of the program: **BATCHEXT**

IV. Mainframe Compile and Testing Operations

TSO22 Compile and Link-Edit the following programs, using the JCL in: DPxxx.RESEARCH.CNTL(MJCOBLNK):

- ABENDRTN: RC_____
- ACCT01PC: RC_____
- RECEIVIP: RC_____
- CNTRLBRK: RC_____
- BATCHRPT: RC_____
- RCPOST: RC_____

TSO23 Resolving COBOL syntax errors - Compile the following program, and resolve all syntax errors:

Save the program after changes

- ERRPLX

TSO24 Compiling programs (a portion of an application) that are affected by a maintenance modification. Enlarge the field: **W50-RETURN-CODE** (part of copybook: **XTK0023F**) from **PIC X(3)**, to **PIC X(5)**. **Recompile any program that uses the copybook, and resolve any syntax errors Save these programs**

TSO25 Batch integration testing – running a job, and reviewing its results - Submit the job: B100EMPL. When it is finished:

- Browse the results and write the return-code from these two steps:
- BR14STEP: _____ GENSTEP1: _____
- Browse the following datasets, and write the value of the data in the 1st byte of the 1st record in the file:
 - TEST.EMPHIRE.RPT: _____
 - TEST.EXTRACT.DATA: _____

TSO26 Testing COBOL Logic – Variable Data Debugging - Re-Submit the job: B100EMPL. But before resubmitting, add the following Debugging code (type DISPLAY statements to monitor all values in the specified variables, and answer the following):

- In BATCHRPT (3000-PROCESS-RECORD & 5000-COMPUTE-VACATION-DAYS paragraphs),

DISPLAY the values in these fields:

- EMPREC-LAST-NAME, EMPREC-DOH-YY, YEARS-EMPLOYED, RPT-VACATION-DAYS
- **Fill in the values for the following:**
- Salt's Years-Employed: _____
- Jones' Vacation-Days: _____
- Duck's EMPREC-DOH-YY: _____
- Smith's Vacation-Days: _____

TSO27 Testing COBOL Logic – Execution (Intra-Module) Path

Verification/Debugging - Re-Submit the job: B100EMPL. But before resubmitting, add the following Debugging code (DISPLAY statements) to prove that certain logic paths are executed:

- In CNTRLBRK determine if the following logic was ever executed by the test run:
 - (paragraph 343-DETAIL-LINE-PRSS): **IF AGED-CODE-SRT-WK IS EQUAL TO 3:**

 - (paragraph 210-PRSS-INPUT-RECORDS): **INPUT-OK AND UNPAID**, but **NAME** not found: _____
 - (paragraph 214-CK-INPUT-DATA): **AGED-CODE-IN > 5 :** _____

- (paragraph 214-CK-INPUT-DATA): **INV-NO-IN-NUM IS not NUMERIC:** _____

V. Mainframe (TSO) – Maintenance Project

TSO28 Instructions

For this project you will use the **Payroll Register** application – (**DPxxx.MAINTAIN**. datasets: - COBOL, COPY and CNTL). These PDSs contain all source code for the Payroll Register application.

There are several input files to this application. All are prepared (Allocated, and loaded with test data).

Task

The field **EMP-FIRST-NAME**, part of the record **EMP-MASTER-RECORD**, that is on the file **DPXXX.EMPVSAM.DATA** must be expanded in the application – from **PIC X(8)** to **PIC X(15)**.

Do the following:

- Run the entire jobstream end-to-end – with the existing source and data files.
- View the return codes from all jobs to ensure that all components are in working order.
- Modify the JCL. Specify the file **DPXXX.EMPVSMF.DATA** in the JCL (assume that the DBA has repro'd the **DPXXX.EMPVSAM.DATA** VSAM file, and expanded **EMP-FIRST-NAME** – *including loading expanded first name test data*)
- Find all elements of the Payroll Register application affected by the expansion of **EMP-FIRST-NAME**
- Expand fields that are affected by the expansion of the data, including:
 - **Programs and Copybooks:**
 - **Elementary fields**
 - **01 or group fields**
 - **FD**
 - **RECORD CONTAINS clause**
 - **01 PIC clause**
 - **Output Files and reports:**

- **JCL**
 - **Change the DSN (data set name) for DPXXX.EMPVSAM.DATA to DPXXX.EMPVSMF.DATA for all JCL jobs in this application**
 - **Change LRECL for all other (output) files affected by the expansion of EMP-FIRST-NAME:**
 - **LRECL – increase by 7 bytes** (do not worry about SPACE parameters)
 - **Do not modify the JCL/DSN's for any other files except for DPXXX.EMPVSMF.DATA**

Note: The order of jobs in this application is:

- **P1CHECKR**
- **P2COMPUT**
- **P3PRINT**
- **P4REGIS**

To verify that your work is complete, after your source changes, run the entire job-stream (all run JCL is set in your .CNTL PDS – which also contains compile and link JCL for your programs) – in the above order, and verify your results by browsing:

- **The job output return-codes**
- **Any SYSOUT displays**
- **Output files created – browse to ensure that the first name field has not been truncated**

Appendix B. Task list two - Revolve/MFE

The following is a list of all PC based MFE/Revolve tasks tested in the study. Each item is numbered sequentially. Some of these items are also marked by a Task Item number, in this case RMFEx or RMFExx, following the sequential number. All tasks bearing a sequence number of this type are a part of a matched pair. The pair is identified by the xx following the RMFE. The item is paired with a corresponding task from the TSO/ISPF test set in **Appendix A**.

VI. Revolve - Specific Research and Analysis tasks

Analysis and Research Questions

RMFE1 List the program(s) that **READ** the file **MFOCUS.SLS135C0**.

RMFE2 List the program(s) that write and/or rewrite the file MFOCUS.ACCT01C0

RMFE3 **List the DATA DIVISION declarations (01-level group variable names) used to access the file: TEST.PRODFILE.DATA.**

RMFE4 List the program(s) that utilize the COPY file: **XYCM004W**

RMFE5 List the modules (program names) in the Calling chain that invoke the subroutine: **ACCT18PA –**

invoked out of the **ACCTSBAT** job.

RMFE6 In a reengineering project to provide online access to data **CNTRLBRK, UTLM046,** and

BATCHEXT will be rewritten to become CICS online programs. There are several COBOL verbs that are not

allowed by CICS. List the program names and line#s of the following verbs: **ACCEPT, SORT, DISPLAY**

RMFE7 There was a problem in the **CALL** from **ACCT19PA** to **ACCT18PA**, which occurs in paragraph:

P4003A-CALL-XQ3012IP.

List the paragraph names from the beginning of the program to the Call statement.

RMFE8 The field **CCR-UPDATE-DATE-JUL** must be increased in size from 5 to 7 bytes.

Which application elements are affected?

List all: Files, Programs, JCL, Program variables, Copybooks (please list field-names) and literals.

RMFE9 What data (list the variable name through which the data is accessed) is contained in bytes:

4 - 5 of the file **FOCUS.ACCT16C2**

VII. MFE Development Operations – Graphical Editing

Using Mainframe Express, accomplish the following. **Save all source changes**

Specific Procedures

RMFE10 Loading a program into the editor: Edit the following programs and insert a single blank line at the bottom of the file:

- ACCT01PC
- ACCT17PA
- CKDIG910
- RECEIVIP
- RCPOST

RMFE11 Scrolling and Navigating within source files. From within the COBOL program - **MSTFILA** –

Type a percent sign at Column 7 in the following source lines:

- The 1st **SELECT** statement

- The first 05-field name in the **SORT-WORK-FILE FD**
- The line that contains the literal value **FEB**
- The **GOBACK** statement line, in **000-MAINLINE**
- The line at the top of file - **IDENTIFICATION DIVISION** statement.
- The bottom of file on the line that contains **AFTER ADVANCING 2.**
- Beginning of Working Storage Section: **WS-OLD-MF-REC.**
- Beginning of Linkage Section: **MSTF-PARMS.**
- **PROD-LINE-FILE-IN – FD**
- First procedural line, at the beginning of paragraph: **100-READ-PROD**

RMFE12 Single Line Insert Operations - From within the COBOL program - MSTFIL0 – insert a single blank line below (after) the following lines (save all changes):

- Bottom of file
- Top of file
- FILE CONTROL
- FD OLD-CLAMS-MF-IN
SELECT PROD-LINE-FILE-IN ASSIGN TO UT-S-PRODFILE.
- 100-INITIALIZE.
- Beginning of Working Storage Section
- Beginning of Linkage Section
- Beginning of Procedure Division
- 110-LOAD-PROD-TABLE paragraph.

46. Delete Operations

RMFE13 Single-Line Delete Operations - From within the COBOL program MSTFIL2 – delete the following single lines (save all changes):

- **SELECT PROD-LINE-FILE-IN ASSIGN TO UT-S-PRODFILE.**
- **DATE-WRITTEN. JULY 1996.**
- **01 OLD-MF-RECORD PIC X(100).**

- **FD OLD-CLAMS-MF-IN**
- **01 WS-OLD-MF-REC.**

RMFE14 Multi-Line Delete Operations - From within the COBOL program MSTFIL3 - Delete the following lines (save all changes):

- All **SELECT ASSIGN** statements
- The entire **REPORT-FILE-OUT FD**
- The entire **WS-PROD-LINE-REC.** 01 group
- The entire **WS-DETAIL-LINE.** 01 group
- All **FILLER** lines under **MONTH-DATA (under MONTH-TABLE-DEFINITION)**
- The entire **100-INITIALIZE** paragraph
- The **READ** statement in the **229-READ-ACTIVITY** paragraph
- The entire **275-CLAM-TOTALS** paragraph
- The **IF NOT EOF-SRT** statement in **280-RETURN-ACTIVITY-RECORD**
- The entire **290-WRITE-NEW-MF** paragraph.

Copy operations

RMFE15 Single-Line Copy - From within the COBOL program MSTFIL4 – copy the following single lines to the top of file(save all changes):

- **SELECT PROD-LINE-FILE-IN ASSIGN TO UT-S-PRODFILE.**
- **DATE-WRITTEN. JULY 1996.**
- **01 OLD-MF-RECORD PIC X(100).**
- **FD OLD-CLAMS-MF-IN**
- **01 WS-OLD-MF-REC.**

RMFE16 Multi-Line Copy - From within the COBOL program MSTFIL5 - Copy the following blocks of lines to the top of file(save all changes):

- All **SELECT ASSIGN** statements
- The entire **REPORT-FILE-OUT FD**

- The entire **WS-PROD-LINE-REC.** 01 group
- The entire **WS-DETAIL-LINE.** 01 group
- All **FILLER** lines under **MONTH-DATA** (under **MONTH-TABLE-DEFINITION**)
- The entire **100-INITIALIZE** paragraph
- The **READ** statement in the **229-READ-ACTIVITY** paragraph
- The entire **275-CLAM-TOTALS** paragraph
- The **IF NOT EOF-SRT** statement in **280-RETURN-ACTIVITY-RECORD**
- The entire **290-WRITE-NEW-MF** paragraph.

Replicate (repeat) Line Operations

RMFE17 Replicate Operations - From within the COBOL program **MSTFIL8** – Duplicate the following single lines

- **SELECT PROD-LINE-FILE-IN ASSIGN TO UT-S-PRODFILE.**
- **DATE-WRITTEN. JULY 1996.**
- **01 OLD-MF-RECORD PIC X(100).**
- **FD OLD-CLAMS-MF-IN**
- **WS-OLD-MF-REC.**

RMFE18 Multi-Line Replicate Operations - From within the COBOL program **MSTFIL9** - Duplicate the following lines

- All **SELECT ASSIGN** statements
- The entire **REPORT-FILE-OUT FD**
- The entire **WS-PROD-LINE-REC.** 01 group
- The entire **WS-DETAIL-LINE.** 01 group
- All **FILLER** lines under **MONTH-DATA** (under **MONTH-TABLE-DEFINITION**)
- The entire **100-INITIALIZE** paragraph
- The **READ** statement in the **229-READ-ACTIVITY** paragraph
- The entire **275-CLAM-TOTALS** paragraph
- The **IF NOT EOF-SRT** statement in **280-RETURN-ACTIVITY-RECORD**
- The entire **290-WRITE-NEW-MF** paragraph.

RMFE19 "FIND" Operation within the context of application research and development activity

From within the COBOL program RCPOST –

Position your cursor on Paragraph: 7710-MFC and find the following:

Write the value clause for the following 88-level data items:

- CUSTOMER-NA_____
- CUSTOMER-DELETE_____

Write the picture clause for the following data items:

- TWO _____
- WH01-REJECT-CODE _____

From the 7711-MORE-MFC paragraph

- What is the last COBOL verb in this paragraph _____

Write the picture clause for the following data items:

- FCTMFC-23-FIELD-NO_____
- WH01-TXN-PHONE-CITY_____

Return to the PERFORM of 7710-MFC:

- If NEW-CUSTOMER, what paragraph is PERFORM'd? _____
- If WK06-TRAN-CODE = 10 what paragraph is executed? _____
- What data value(s) in WK06-TRAN-CODE cause 7720-NAME-CHANGE to be executed? _____

RMFE20 Discretionary and Global "Change" Operations - From within, RCPOSTB – Change the following:

- Replace the 1st occurrence of WA01-CUSTOMER-BAL with WS01-WORK-FLD-NUM
- Replace all occurrences of NEW-RMST2-CUSTOMER-NO with NEW-RMS6TH-FLD-RED

- Replace all occurrences of 6000-CLNT-TOTALS with 5980-MAX-PERFORM-INTRST
- Replace the last occurrence of WS01-PREVIOUS-RECORD with WS88-REC-EXCPTN
- Replace the 3rd and 6th occurrences of LW40-ABEND-CODE with WKS-ABEND-RTN
- Replace all occurrences of NEW-RMST2-RECEIVABLES-BALANCE with WS-STOP-PYMNT
- Replace the 1st & last occurrence of LW20-INIT-DATE-PARMS with LW40-ABEND-CODE
- Replace every other occurrence of TRANS-FILE-IN with WS-DATA-RECORD-IN
- Replace all occurrences of NEW-CUSTOMER-MASTER-REC with OLD-CUST-MASTER-REC
- Replace the 1st occurrence of WA01-WORK-BALANCE with – WS-INTRNAL-BALANCE-99

RMFE21 Copy multiple lines from one file to another

VIII. Compile and Testing Operations

RMFE22 Compiling single programs - Compile the following programs:

- ABENDRTN: RC_____
- ACCT01PC: RC_____
- RECEIVIP: RC_____
- CNTRLBRK: RC_____
- BATCHRPT: RC_____
- RCPOST: RC_____

RMFE23 Resolving COBOL syntax errors - Compile the following program, and resolve all syntax errors:

- **BOMB00**

RMFE24 Compiling programs (a portion of an application) that are affected by a maintenance modification - Enlarge the field: **TRX-TRANS-CT** (part of copybook: **XPRD045F**) from **PIC 9(3)**, to **PIC 9(5)**. **Recompile any program that uses the copybook, and resolve all syntax errors** (*note – do not worry about the impact this field change might have on other variables – simply compile and link all programs – and only those programs - that are affected by the change to the Copybook*).

RMFE25 Batch integration testing – running a job, and reviewing it's results

Submit the job: B100EMPL. When it is finished:

- Browse the results and write the return-code from each step:
- RPTSTEP: _____ CNTRLBRK: _____ GENSTEP2: _____
- Browse the following datasets, and write the data value in the 2nd byte of the 1st record
 - TEST.EMPHIRE.RPT: _____
 - TEST.EXTRACT.DATA: _____

RMFE26 Testing COBOL Logic – Variable Data Debugging

Re-Submit the job: B100EMPL. During Debugging (Animation) set breakpoints, double-Click fields ... Add-to-list, in order to determine the answers to the following):

- **In BATCHRPT (3000-PROCESS-RECORD & 5000-COMPUTE-VACATION-DAYS paragraph) monitor these fields:**
 - **EMPREC-LAST-NAME, EMPREC-DOH-YY, YEARS-EMPLOYED, RPT-VACATION-DAYS**
- Fill in the values for the following:
 - Petter's Years-Employed: _____
 - Davies' Vacation-Days: _____
 - Clinton's EMPREC-DOH-YY: _____
 - Tucker's Vacation-Days: _____

RMFE27 Testing COBOL Logic – Execution (Intra-Module) Path

Verification/Debugging - Re-Submit the job: B100EMPL. But before resubmitting, add the following Debugging code:

- **In CNTRLBRK determine if the following logic is ever executed:**
 - (paragraph **343-DETAIL-LINE-PRSS**): **IF AGED-CODE-SRT-WK IS EQUAL TO 3:**

 - (paragraph **210-PRSS-INPUT-RECORDS**): **INPUT-OK** and **UNPAID**, and **NAME** found: _____

- (paragraph **214-CK-INPUT-DATA**): **AGED-CODE-IN <= 5**: _____
- (paragraph **214-CK-INPUT-DATA**): **INV-NO-IN-NUM NUMERIC**: _____

IX. Mainframe Express – Maintenance Project

RMFE28 For this project you will use the **Payroll Register** application (the Payroll project). See the BenchMark supervisor if you need assistance finding and loading this application – for both MFE and Revolve)

The field **EMP-LAST-NAME**, part of the record **EMP-MASTER-RECORD**, that is on the file **DPXXX.EMPVSAM.DATA** must be expanded in the application – from **PIC X(14)** to **PIC X(20)**.

Do the following:

- Run the entire jobstream end-to-end. View the return codes from all jobs to ensure that all components are in working order.
- Modify the JCL. Specify the file **DPXXX.EMPVSML.DATA** in the JCL (the DBA has repro'd the file, and expanded **EMP-LAST-NAME** in this input VSAM file)
- Find all elements of the Payroll Register application and, expand fields that are affected by the expansion of the data, including:
 - **Programs and Copybooks:**
 - **Elementary fields**
 - **01 or group fields**
 - **FD**
 - **RECORD CONTAINS clause**
 - **01 PIC clause**
 - **Output Files and Sysout reports:**
 - **JCL**
 - **Change the DSN (data set name) for DPXXX.EMPVSAM.DATA to DPXXX.EMPVSML.DATA for all jobs in this application**

- **Change LRECL for all other files affected by the expansion of EMP-FIRST-NAME as follows:**
- **LRECL – increase by 6 bytes**
- **Do not modify the JCL/DSN's for any other files except for DPXXX.EMPVML.DATA**

Note: The order of jobs in this application is:

- **P1CHECKR**
- **P2COMPUT**
- **P3PRINT**
- **P4REGIS**

To verify that your work is complete, after your source changes, run the entire job-stream from MFE (all run JCL is set in your project) – in the above order, and verify your results by browsing:

- **The job output return-codes**
- **Any SYSOUT displays**
- **Output files created – browse to ensure that the JobCode field has not been truncated**

Revolve steps (from within Revolve)

Open the Maintain project

Impact Trace EMP-LAST-NAME

- **Pulldown Tools, Impact Analysis**
- **Type in: EMP-LAST-NAME**
- **Click Match**
- **Click Impact Tracing**

Setup Impact Viewing Options - and filter un-needed information

- **Double-Click the Group 1: folder**
- **Select (highlight) the buffer FILLER entries and click the Red X (remove) them**
- **Select (highlight) the copybook and req_entity_constant_spaces entries and Red X (remove them)**
- **Click View Source Code**

Expand all PIC clauses and file FDs

- Click each of the elementary item folders in the top half of the window to position your cursor on the source to be changed
 - Edit the PIC clause in the source – add 6 bytes to each field length
- Click each of the buffers (group fields) in the top half of the window to position your cursor on the source to be changed
 - Edit the PIC clause in the source – add 6 bytes to each field

Expand the RECORD CONTAINS clause in the FD

- Double-click each File name
- From the upper (Impact Trace) window Click the first line under the expanded Filename folder (**Component programname.cbl accesses**)
- From lower window (source code view), scroll down in the program source until you find the FD for the file
- Expand the RECORD CONTAINS clause (add 6 bytes)
- Double-click the file name again, to close contract (close) the file folder

Close the View Source Code window – (click the **View Source Code** icon)

Change the JCL and expand the LRECL. For each of the files in the report, do the following:

- Right-Click each **file folder** in the top half of the window, and select **Information**
- Highlight the filename in the **Data Files Browser** window, and **Expand Completely**
- Click **View Source Code**
- For each JCL entry:
 - If the file disposition is: **DISP=NEW**, do the following:
 - Increase the **LRECL** by 6 bytes
 - If the file is **DPXXX.EMPVSAM.DATA** change the DSN (**DPXXX.EMPVSML.DATA**)

Close the Impact Trace window and click: **Yes to all** - at the Save File prompt

MFE Steps (from within Mainframe Express)

Open the Maintain project

- Rebuild the project
- Build menu
- Select Build
- Run the complete JCL job stream

Recall: The order of jobs in this application is:

- **P1CHECKR**
- **P2COMPUT**
- **P3PRINT**
- **P4REGIS**

To verify that your work is complete, after your source changes, run the entire job-stream from MFE (all run JCL is set in your project) – in the above order, and verify your results by browsing:

- **The job output return-codes**
- **Any SYSOUT displays**
- **Output files created – browse to ensure that the JobCode field has not been truncated**

If you encounter any problems or discrepancies in the output, set break points at the MOVE statements, Step and Monitor your execution at the source level using Animator.

Appendix C. Categorization of Mainframe and PC Tasks

This appendix contains a categorization and description of the types of tasks listed in Appendices A and B.

II Mainframe Research and Analysis Tasks

Before software maintenance can be accomplished the target of the maintenance must first be found.

The following group of tasks measures the time required to execute those activities used to find and manipulate specific instances of data required to accomplish common maintenance tasks.

TSO1 through TSO9

III. Development Operations - ISPF Editing Tasks

Once the research and analysis tasks are completed, the actual software maintenance is accomplished through editing of source programs and data.

The following group of tasks measures the time needed to execute edit operations commonly used to accomplish software maintenance.

TSO10 through TSO21

IV. Mainframe Compile and Testing Operations

After the maintenance solution has been implemented, through editing operations, it must be tested and errors removed.

The following tasks measure the time necessary to accomplish activities related to compiling of maintenance changes.

TSO22 through TSO24

The following tasks measure the time necessary to accomplish activities related to testing of maintenance changes.

TSO25 through TSO27

V. Mainframe Maintenance Project

The following task is an integrated activity, which measures the time necessary to accomplish the three previously, discussed steps of research, editing and compilation and testing.

TSO28

VI. Revolve Research and Analysis Tasks

Before software maintenance can be accomplished the target of the maintenance must first be found.

The following group of tasks measures the time required to execute those activities used to find and manipulate specific instances of data required to accomplish common maintenance tasks.

RMFE1 through RMFE9

VII. MFE Development Operations - Graphical Editing Tasks

Once the research and analysis tasks are completed, the actual software maintenance is accomplished through editing of source programs and data.

The following group of tasks measures the time needed to execute edit operations commonly used to accomplish software maintenance.

RMFE10 through RMFE21

VIII. MFE Compile and Testing Operations

After the maintenance solution has been implemented, through editing operations, it must be tested and errors removed.

The following tasks measure the time necessary to accomplish activities related to compiling of maintenance changes.

RMFE 22 through RMFE 24

The following tasks measure the time necessary to accomplish activities related to testing of maintenance changes.

RMFE 25 through RMFE 27

IX. Mainframe Express Maintenance Project

The following task is an integrated activity, which measures the time necessary to accomplish the three previously, discussed steps of research, editing and compilation and testing.

RMFE28

Appendix D. Demographics of Participants Included in Final Results

Name	Yrs Current Job	Yrs Prog Exp	Yrs Exp COBOL	Yrs Exp Windows	Yrs Exp Windows Prod Tools	Yrs Exp MVS	Yrs Exp TSO/ISPF
S1	1	12	12	11	11	12	12
S2	8	20	20	8	8	20	20
S3	3	10	3	10	10	10	3
S4	2	2	2	8	8	2	2
S5	10	17	17	4	4	17	17
S6	1	20	10	5	5	12	12
S7	10	32	15	5	5	12	12
S8	1	14	10	3	3	14	14

Appendix E. Elapsed Time Results - TSO tasks

Elapsed Time (minutes) TSO - All SS

Task	Beginner					Expert			Avg	SD	Min	Max
	S2	S3	S4	S5	S6	S7	S8	S1				
TSO1	6	11	10	5	2	14	23	9	10.14	6.96	2.00	23.00
TSO2	5	2	12	7	3	5	5	4	5.57	3.26	2.00	12.00
TSO3	4	4	13	5	4	2	5	6	5.29	3.55	2.00	13.00
TSO4	10	5	5	1		3	2	11	4.33	3.20	1.00	10.00
TSO5	4	6	1	1		14	4	2	5.00	4.82	1.00	14.00
TSO6	7	54	9	5	12	5	5	13	13.86	17.89	5.00	54.00
TSO7	8	10	14	6	8	10	12	8	9.71	2.69	6.00	14.00
TSO8	7	25		5	11	30	33	35	18.50	12.29	5.00	33.00
TSO9		13	1	5		3	8	3	6.00	4.69	1.00	13.00
TSO10	4	23	4	2	2	2	2	5	5.57	7.74	2.00	23.00
TSO11	5	15	13	7	10	4	5	4	8.43	4.31	4.00	15.00
TSO12	3	25	6	4	2	2	3	6	6.43	8.30	2.00	25.00
TSO13	1	2	3	7	2	3	1	1	2.71	2.06	1.00	7.00
TSO14	4	5	5	3	5	3	5	2	4.29	0.95	3.00	5.00
TSO15	3	2	3	1	2	2	1	2	2.00	0.82	1.00	3.00
TSO16	5	8	5	3	12	3	2	6	5.43	3.51	2.00	12.00
TSO17	1	3	2	1	1	1	3	1	1.71	0.95	1.00	3.00
TSO18	4	7	5	2	4	2	2	5	3.71	1.89	2.00	7.00
TSO20	14	14		11		13	15	20	13.40	1.52	11.00	15.00
TSO21	4	2	2	1		2	5	7	2.67	1.51	1.00	5.00
TSO22		21	15	13		18	17	13	16.80	3.03	13.00	21.00
TSO23						20	20		20.00	0.00	20.00	20.00
TSO24	15	18		5		7	8	21	10.60	5.59	5.00	18.00
TSO25	3	7	14	2		5	5	5	6.00	4.29	2.00	14.00
TSO26	10	32	35	9		8	7	14	16.83	12.98	7.00	35.00
TSO27	10	19	25	8		6		11	13.60	8.08	6.00	25.00
TSO28	10			35				23	22.50	17.68	10.00	35.00
Total	147	333	202	154	80	187	198	237	185.86	77.23	80.00	333.00
Average Total Time									185.86			

Key: Boldfaced Elapsed Time numbers indicate less than the number shown (e.g. 1 is <1)

Appendix F. Elapsed Time Results - Revolve/MFE tasks

**Elapsed Time (minutes)
Revolve/MFE - All SS**

task	Beginner					Expert User			Avg	SD	Min	Max	
	S2	S3	S4	S5	S6	S7	S8	S1					
RMFE1	1	3	1	2		2.50	1.15	3	1.78	0.86	1.00	3.00	
RMFE2	3	2	1	6		1.50	1.00	1	2.42	1.91	1.00	6.00	
RMFE3	4	5	2	4		1.58	3.00	4	3.26	1.31	1.58	5.00	
RMFE4	3	4	2	2		1.50	2.00	1	2.42	0.92	1.50	4.00	
RMFE5	6	3	2	5		1.75	4.00	5	3.63	1.69	1.75	6.00	
RMFE6	6	9	12	2		3.00	4.00	4	6.00	3.85	2.00	12.00	
RMFE7	2	2	6			1.25	3.00	2	2.85	1.87	1.25	6.00	
RMFE8	7	7	11	7		5.25	6.00	10	7.21	1.99	5.25	11.00	
RMFE9	4	2	3	3		2.00	1.00	2	2.50	1.05	1.00	4.00	
RMFE10	4	10	2	3		1.00	2.00	6	3.67	3.27	1.00	10.00	
RMFE11	3	13	27	5		2.00	5.00	4	9.17	9.56	2.00	27.00	
RMFE12	5	13	5	6		1.00	2.00	1	5.33	4.23	1.00	13.00	
RMFE13	1	4	2	4		1.00	4.00	1	2.67	1.51	1.00	4.00	
RMFE14	4	6	10	15		1.00	3.00	2	6.50	5.17	1.00	15.00	
RMFE15	4	5	1	14		1.00	2.00	3	4.50	4.93	1.00	14.00	
RMFE16	5	11	6	13		3.00	5.00	5	7.17	3.92	3.00	13.00	
RMFE17	1	2	2	6		1.00	1.00	2	2.17	1.94	1.00	6.00	
RMFE18	7	8	5	6		2.00	4.00	4	5.33	2.16	2.00	8.00	
RMFE20	15	22	15			2.00	7.00	9	12.20	7.79	2.00	22.00	
RMFE21	2	2	3	2		1.00	2.00	1	2.00	0.63	1.00	3.00	
RMFE22	2	8	3	1		2.00	1.00	2	2.83	2.64	1.00	8.00	
RMFE23	5			6		2.00	3.00	7	4.00	1.83	2.00	6.00	
RMFE24	2	8	2	2			4.00	2	3.60	2.61	2.00	8.00	
RMFE25	6	14	5	2		2.00	3.00	4	5.33	4.55	2.00	14.00	
RMFE26	5	7	8	8		4.00	5.00	7	6.17	1.72	4.00	8.00	
RMFE27	20	18				8.00	15.00	17	15.25	5.25	8.00	20.00	
RMFE28	8		5					24	6.50	2.12	5.00	8.00	
Total	135.00	188.00	141.00	124.00		54.33	93.15	133.00	124.07	41.62	54.33	188.00	
Average Total Time									124.07				

Key: Boldfaced Elapsed Time numbers indicate less than the number shown (e.g. 1 is <1)

Appendix G. Paired Samples Test Result for All Subjects

		Paired Samples Test					
		Paired Differences					
		Mean	Std. Deviation	Std. Error Mean	t	df	Sig. (2-tailed)
Pair 1	TSO1 - MFE1	9.1229	6.2331	2.3559	3.902	6	.008
Pair 2	TSO2 - MFE2	3.5000	3.5940	1.3584	2.577	6	.042
Pair 3	TSO3 - MFE3	2.2029	4.0251	1.5213	1.448	6	.198
Pair 4	TSO4 - MFE4	3.0714	4.0045	1.5135	2.049	6	.089
Pair 5	TSO5 - MFE5	.7500	5.5547	2.0995	.357	6	.733
Pair 6	TSO6 - MFE6	8.2857	16.5803	6.2667	1.322	6	.234
Pair 7	TSO7 - MFE7	7.6250	1.3205	.5391	14.144	6	.000
Pair 8	TSO8 - MFE8	15.4583	13.1210	5.3566	2.886	6	.034
Pair 9	TSO9 - MFE9	3.333	4.7610	1.9437	1.715	6	.147
Pair 10	TSO10 - MFE10	2.000	4.9666	1.8772	1.065	6	.328
Pair 11	TSO11 - MFE11	-.8571	5.8716	2.2193	-.386	6	.713
Pair 12	TSO12 - MFE12	2.2857	4.8892	1.8480	1.237	6	.262
Pair 13	TSO13 - MFE13	.1429	2.1157	.7997	.179	6	.864
Pair 14	TSO14 - MFE14	-2.0000	5.0000	1.8898	-1.058	6	.331
Pair 15	TSO15 - MFE15	-2.2857	4.9905	1.8862	-1.212	6	.271
Pair 16	TSO16 - MFE16	-2.2857	3.7289	1.4094	-1.622	6	.156
Pair 17	TSO17 - MFE17	-.4286	2.2254	.8411	-.510	6	.629
Pair 18	TSO18 - MFE18	-1.2857	1.7995	.6801	-1.890	6	.108
Pair 19	TSO19 - MFE19						
Pair 20	TSO20 - MFE20	4.2000	8.4083	3.7603	1.117	4	.327
Pair 21	TSO21 - MFE21	1.4286	2.5071	.9476	1.508	6	.182
Pair 22	TSO22 - MFE22	13.3333	2.1602	.8819	15.119	5	.000
Pair 23	TSO23 - MFE23	17.5000	.7071	.5000	35.000	1	.018
Pair 24	TSO24 - MFE24	9.8000	6.6106	2.9563	3.315	4	.030
Pair 25	TSO25 - MFE25	.7143	4.9905	1.8862	.379	6	.718
Pair 26	TSO26 - MFE26	10.1429	11.0216	4.1658	2.435	6	.051
Pair 27	TSO27 - MFE27	-4.2500	4.7871	2.3936	-1.776	3	.174
Pair 28	TSO28 - MFE28	.5000	2.1213	1.5000	.333	1	.795

Significance at .05 or less

Micro Focus

Micro Focus – Unlocking the Value of Legacy™

Micro Focus is a leading provider of legacy development and deployment software for enterprise platforms. Micro Focus enables organizations to reduce costs and increase agility with minimal risk by reusing their legacy applications with contemporary architectures and Web services. Founded in 1976, Micro Focus is a global company with principal offices in the United Kingdom, United States and Japan. For more information, visit www.microfocus.com.

Micro Focus Worldwide

Australia.....	1800 632 626
Austria.....	0800 293 535
Belgium.....	0800 11 282
Canada.....	905 824 7397
France.....	0800 916 564
Germany.....	0800 182 5443
Ireland.....	353 1 469 3121
Italy.....	800 784 420
Japan.....	81 3 5793 8550
Luxembourg.....	800 23743
Netherlands.....	0800 022 8562
Norway.....	47 22 91 07 20
Switzerland.....	0800 564 247
United Kingdom.....	0800 328 4967
United States.....	877 772 4450
Other Countries.....	44 1635 32646

© 2005 Micro Focus. All Rights Reserved.

WPIMSM0505-US