AN EMPIRICAL STUDY OF FUNCTION POINTS ANALYSIS RELIABILITY

by

Michael J. Connolley

B.S. Civil Engineering, Syracuse University
(1980)

Submitted to the MIT Sloan School of Management
in Partial Fulfillment of the Requirements of the Degree of
Master of Science in Management

at the

Massachusetts Institute of Technology
June 4, 1990

Signature of Author ______________________________
MIT Sloan School of Management
May 16, 1990

Certified by ______________________________
Chris F. Kemerer
Douglas Drane Career Development Assistant Professor of
Information Technology and Management
Thesis Supervisor

Accepted by ______________________________
Jeffery A. Barks
Associate Dean, Master's and Bachelor's Program

AN EMPIRICAL STUDY OF FUNCTION POINTS ANALYSIS RELIABILITY

by
Michael J. Connolley

Submitted to the MIT Sloan School of Management
in Partial Fulfillment of the Requirements of the Degree of
Master of Science in Management

ABSTRACT

The Function Points methodology was developed by Allan Albrecht to help measure the size of software programs. The methodology quantifies the amount of functionality provided by an application and expresses it in a metric called the Function Point. Such size measurements are needed to quantify development and maintenance productivity, and to estimate development work effort. The Function Points methodology has been adopted by over 500 major corporations and is rapidly growing in popularity. Numerous variants of the Function Points methodology have been proposed.

Despite the growing use of Function Points, these is a widespread perception that the Function Points metric is not reliable; that different people estimating the same application will get much different results.

This study looks at the reliability of the Function Points metric in two ways: first it quantifies the reliability of Function Point counts made by raters using exactly the same methodology; and second, it quantifies the reliability of counts made using two different Function Points variants. Furthermore, the study examines some of the factors which could affect reliability.

Thesis Supervisor: Chris F. Kemerer
Title: Douglas Drane Career Development Assistant Professor of Information Technology and Management

ACKNOWLEDGEMENTS

The author gratefully acknowledges that this study would not have been possible without the cooperation of the International Function Points Users Group (IFPUG). IFPUG not only funded the study, but IFPUG's membership donated their time and expertise to provide the data upon which the study is based. Special recognition is due to Andrew Belden and Benjamin Porter of the IFPUG Counting Practices Committee for both their administrative aid and for their very helpful suggestions about the study.

Additional acknowledgment is due to those who provided feedback on the first draft of the data collection instrument. These people are A. Albrecht, J. Cooprider, B. Dreger, R. Jeffery, C. Jones, T. Mukhopadhyay, D. Reifer, H. Rubin, E. Rudolph, C. Symons, and N. Venkatraman.

Any remaining errors, of course, are the responsibility of the author.

TABLE OF CONTENTS

1. INTRODUCTION

The Need for Software Metrics

Software is taking on an increasingly important role in business. As Pressman states, "In essence software is often the key factor that differentiates. The completeness and timeliness of information provided by software (and related data bases) differentiates one company from its competitors" [Pressman, 1987].

Unfortunately, despite its importance, software development is often poorly managed. Pressman describes a software development crises,

> "The software crises is characterized by many problems, but managers responsible for software development concentrate on the 'bottom line' issues: (1) schedule and cost estimates are often grossly inaccurate; (2) the 'productivity' of software people hasn't kept pace with the demand for their services; and (3) the quality of software is sometimes less than adequate. Cost overruns of an order of magnitude have been experienced. Schedules slip by months or years. Little has been done to improve the productivity of software practitioners. Error rates for new programs cause customer dissatisfaction and lack of confidence." [Pressman, 1987]

Software metrics and software estimating methodologies have been developed to help alleviate the software development crisis. A software metric is a unit of measure for the size of a piece of software. Software estimating methodologies are techniques for calculating these software metrics.

"When comprehensive software metrics are available, estimates can be made with greater assurance, schedules can be established to avoid past difficulties, and overall risk can be reduced" [Pressman, 1987]. More specifically, software metrics help with the management of specific software development projects, and they help with improving the software development and maintenance processes. For software development, software metrics can be used for manpower planning, cost estimation, scheduling, progress measurement, and assessing the impact of changes. For improving the development and maintenance processes, software metrics can help quantify changes in productivity, overhead, and quality (i.e. lack of bugs) [Pressman, 1987].

Description of the Function Points Metric

The Function Points (FP) metric was developed by Allan Albrecht [Albrecht & Gaffney,1983]. It is a measure of the functionality a piece of software provides the user. It is applicable to management information systems rather than real time or computation-intensive applications. A typical application has 300 to 700 Function Points [Rudolph, 1990; Albrecht 1990]. Albrecht described the methodology as follows:

> "The function value delivered to the user, as viewed through a software program's external attributes, is measured as a dimensionless number called Function Points. ... The number of Function Points attributed to a unit of software is determined by counting and weighting the number of external user inputs, inquiries, outputs, and master files to be delivered by the software. ... The objective was to develop a relative measure of function value delivered to the user that was independent of the particular technology or approach used" [Albrecht, 1979].

Prior to the introduction of the Function Points methodology, the primary metric for software size was lines of code. There are significant problems with this metric. First, there is no standard definition of what to include in a lines of code count. For example, comment lines may or may not be counted. Jones lists eleven counting variations [Jones, 1986]. The second problem is that a lines of code count is technology dependent. That is, the same program, written in two different languages, can have radically different counts. This technology dependence prevents any kind of cross-language productivity comparisons.

Function Point counts are used to measure changes in development and maintenance costs, quality, and productivity. Additionally, correlations can be developed which relate Function Point counts with development and maintenance work-effort. Once these correlations have been established, the Function Points methodology can be used as an estimating tool.

The Function Points metric has four principle strengths. These are:

- It is technology independent. That is, the same application will have the same number of Function Points regardless of the language the application is written in.
- Function Point counts can be made very early in development.
- Function Point counts of planned applications require relatively little preparation time.

- Function Points analysis works. Its usefulness in estimating work-hours and in quantifying productivity improvements has been well documented [Albrecht & Gaffney, 1983; Behrens, 1983; Kemerer, 1987].

The Function Points Methodology

The Function Points metric measures the application in terms of what is delivered, not how it is delivered. Only user-requested and visible components are counted. These components are categorized as either Data or Transactional Function Types. Data Function Types evaluate the functionality provided the user by the data storage requirements of an application. Transactional Function Types evaluate the functionality provided the user by the processing requirements of an application. The Function Types are:

Data Function Types
- *Internal Logical Files.* This Function Type describes data which resides internal to an application's boundary and reflects data storage functionality provided to the user. Internal Logical Files must be maintained and utilized by the application.
- *External Interface Files.* This Function Type describes data which resides external to an application's boundary and reflects the functionality provided by the application through the use of data maintained by other applications.

Transactional Function Types
- *External Inputs.* This Function Type reflects the functionality provided the user for the receipt of and maintenance of data in Internal Logical Files.
- *External Outputs.* This Function Type reflects the functionality provided the user for output generated by the application.
- *External Inquiries.* This Function Type reflects the functionality provided the user for on-line queries of Internal Logical Files or External Interface Files.

While both Internal Logical Files and External Interface Files contain the word "file" in their title, they are not files in the traditional data processing sense of the word. In this case, file refers to a logically related group of data and not the physical implementation.

Preparing a Function Point count is a four step process.

The First Step. The rater (i.e. the person making the count) counts the number of components of each Function Type and complexity. These are referred to as the component counts (CCs). Components can have Low, Average, or High complexity.

The complexity rating depends upon the number of data elements, record elements, and logical files accessed.

The Second Step. The rater assigns Function Point Values (FPVs) to each CC and sums the FPVs to produce a Function Type Function Point Count (FTFPC). These FPVs range from 3 to 15 and are based on the Function Type and complexity rating. For example, a printed report is an External Output Function Type; and, if it has 5 data elements which are drawn from 2 logical files, it would be of average complexity and have a FPV of 5. Mathematically,

$$FTFPC = \sum_{i=1}^{3}(FPV_i * CC_i)$$

where i represents the three complexity ratings.

The Third Step. The rater sums the FTFPCs to get the Unadjusted Function Point Count (UFPC). Mathematically,

$$UFPC = \sum_{i=1}^{5} FTFPC_i$$

where i represents the five Function Types.

The Fourth Step. The UFPC is then weighted by the Value Adjustment Factor (VAF) to produce the Adjusted Function Point Count (AFPC). The VAF can range from 0.65 to 1.35. It is developed by estimating the relative influence of 14 General System Characteristics (GSCs). A factor that is not present or which has no influence has a weighting of zero, and a factor which has a strong influence throughout the application has a weighting of five. These General Systems Characteristics adjust the Function Count to reflect the overall complexity of the application. The formulas for using the General System Characteristics are:

$$VAF = (GSC * 0.01) + 0.65$$
$$AFPC = UFPC * VAF$$

The General System Characteristics are:

1. Data Communication
2. Distributed functions
3. Performance
4. Heavily used configuration
5. Transaction rate
6. On-line data entry

7. End user efficiency
8. On-line update
9. Complex processing
10. Reusability
11. Installation ease
12. Operational ease
13. Multiple sites
14. Facilitate change [Albrecht & Gaffney, 1983]

Overview of the History of Function Points

Allan Albrecht developed the Function Points methodology while working for IBM in their DP Services organization. He first presented the methodology at an IBM Applications Development Symposium in 1979 [Albrecht, 1979]. In 1982, GUIDE, an association of IBM's customers, began a project "To refine the existing definitions of Function Points so that independent companies / divisions / individuals would get the same results when analyzing the same project" [Zwanzig, 1984]. The project resulted in a Handbook for Estimating Using Function Points in 1984. Prior to publication of the manual, the Function Points methodology was made public in an article by Albrecht and Gaffney [Albrecht & Gaffney, 1983].

In 1986, the International Function Points Users Group (IFPUG) was formed to support and further standardize the Function Points methodology. IFPUG currently has over 200 members and is growing. It has recently released a new counting practices manual [Sprouls, 1990]. In parallel with IFPUG's efforts to standardize Function Points, numerous variants of the methodology have been proposed. A number of these will be described in the Literature Review section.

Dreger summarizes the current status of the Function Points methodology as follows:

> "As of 1988, some 500 major corporations throughout the world are using Function Points and the number of individual projects measured with Function Points exceeds 25,000. The rate of growth of Function Points usage has been doubling each year, and the methodology is rapidly becoming the de facto world-wide standard for measuring information systems" [Dreger, 1989].

Criticisms

Despite the growing use of Function Points, there is a widespread perception that Function Points are not reliable. "Opponents claim that the [Function Points] method requires some 'slight of hand' in that the computation is based on subjective, rather than objective, data" [Pressman, 1987]. Additionally, Boehm describes the definitions of the function types as "ambiguous" [Boehm, 1987].

Scope and Motivation of this Study

Two dimensions for assessing the quality of any metric are accuracy and reliability. The accuracy of a software metric refers to how well it predicts development and maintenance work-effort. As will be discussed in the Literature Review Section, the accuracy of the Function Points metric is well documented.

Reliability is "the extent to which the same observational procedure in the same context yields the same information" [Kirk & Miller, 1989]. Two sub-issues are:

- Inter-Rater Reliability. This refers to the degree of consensus which different raters will have when analyzing the same application using the same methodology.
- Inter-Method Reliability. This refers to the degree of consensus when different methodologies are used.

As will be discussed in the Literature Review section, there is disagreement over the Inter-Rater Reliability of the Function Points metric.

The Inter-Method reliability of the metric is also a contentious issue. There are numerous variants of the Function Points methodology, but the inter-method reliability of counts made using these variants is relatively unstudied.

Both the Inter-Rater and the Inter-Method aspects of the reliability of the Function Points methodology require additional research. This study addresses that need. The study quantifies the reliability of Function Points and explores the causes of discrepancies.

2. LITERATURE REVIEW

Accuracy of the Function Points Metric

There is considerable discussion in the literature on the accuracy of the Function Points metric. The most important of these articles are summarized here. Although reliability rather than accuracy is the subject of this study, these articles are summarized here in order to provide the background needed to understand the reliability results.

In 1983, Albrecht and Gaffney published a two-part validation of the Function Points metric [Albrecht & Gaffney, 1983]. The first part of the validation involved 24 applications develop by IBM's DP Services Department. Albrecht and Gaffney correlated three software size metrics (Function Points, Halstead's software science formulas, and source lines of code (SLOC)) with the work-hours required to develop the applications. The three metrics were found to be equivalent. All had correlation coefficients of at least .86 and an average relative error[1] of less than 32.3%.

In the second part of Albrecht and Gaffney's validation, the correlation between Function Points and SLOC was checked against another set of data consisting of 17 applications. The correlation was greater than .94 and the average relative error was less than 18.6%. Albrecht and Gaffney also found that the number of lines of code per Function Point varied with the programming language.

In 1983, Behrens used data from 24 development projects to demonstrate that Function Points were sufficiently accurate to quantify productivity improvement. The results are:

Year	Cost Range (Work-Hrs/FP)	Mean Cost (W-Hr/FP)	Productivity Index[2]
1980	9.7-47.9	18.3	1.00
1981	2.1-23.4	9.4	0.74

Based upon these data, Behrens concluded that productivity had increased by 26% [Behrens, 1983].

[1](E-A)/A where E is estimated value and A is actual value

[2]A/E where A is actual value and E is estimated value

In 1987, Kemerer tested the accuracy of four models for estimating software development work-effort; two that use Source Lines of Code (SLOC) as an input and two that measure function. The models were: COCOMO, SLIM, the Albrecht 1983 Function Points methodology, and ESTIMACS.

Using 15 projects, Kemerer compared the actual work-months required to those estimated by the four models. The accuracy of the models was measured using Magnitude of Relative Error (MRE)[3] and the adjusted R^2 from a regression which related predicted man-months to actual man-months. (For the Function Points methodology, Function Points were related to actual man-months). The results are:

Model	MRE	Adjusted R^2
Slim	772%	87.8%
Cocomo	581%-610%	52.5% - 68.0%
Function Pts	103%	55.3%
Estimacs	85%	13.4%

Kemerer concluded that the function-based models are more accurate than the SLOC-based models. He noted, however, that there were wide swings in how accurate the models were for a given project [Kemerer, 1987].

In summary, research on Function Points has generally agreed that the Function Points metric can be correlated with development work-effort and lines of code (although there are different correlations for different programming languages), and that the metric is sufficiently accurate to measure changes in productivity.

Inter-Rater Reliability of the Function Points Metric

This section summarizes the available information on the inter-rater reliability of the Function Points metric.

Two anecdotal sources claim that the inter-rater reliability of the metric is quite high. In his textbook, Dreger claims that Function Points estimates are "within 10% for existing systems and 15-20% for planned systems" [Dreger, 1989]. Dreger does not describe the source of the figures, nor does he describe how he is measuring reliability. Dreger also makes the observation that "The main source of error in Function Points Analysis, particularly in early development stages, in incomplete or inaccurate

[3] $| (MM_{est} - MM_{act}) / MM_{act} |$ where MM is man-months

specifications and oversight by the counter, NOT the FPA methodology itself! [-author's emphasis]" [Dreger, 1989]

Rudolph reports that, based on a substantial database of measurements taken during the past six years, the coefficient of variation (i.e. standard deviation divided by mean) should not exceed 10% when measuring fully defined applications. Rudolph also believes reliability has improved recently. He attributes the improvement, in part, to the more explicit counting rules provided by the 1984 IFPUG Release 2.0 guidelines [Rudolph, 1990].

Contrasting Dreger's and Rudolph's claims of high reliability are the previously mentioned criticisms that Function Points Analysis requires slight of hand, and that its rules are ambiguous.

Turning to more substantial information sources, some published studies indicate relatively poor reliability.

In 1983, Rudolph described an earlier GUIDE exercise where about 20 individuals made Function Point counts based on a requirements document, the results of which were within 30% of the average. These individuals had only vague counting rules. [Rudolph, 1983].

In 1990, Low and Jeffery studied the reliability of Function Point counts made using common program specifications. They used two different program specifications, each of which were analyzed by 22 experienced raters. The applications analyzed were small, having 58 and 40 function points. Low and Jeffery found that for project 1, the standard deviation was within 45.5% of the mean, and for project 2, standard deviation was within 33.8% of the mean. Low and Jeffery attributed the poor inter-rater reliability to differences in interpretation of the counting rules or the program specification (i.e. the inherent subjectivity of FP analysis). No systematic errors or specific problem areas were identified.

Low and Jeffery also studied the differences in reliability of counts made by experienced versus inexperienced raters. They concluded that Function Point counts show decreasing reliability among the following groups: Experienced Function Points raters, people who were experienced in information systems but inexperienced in

Function Points counting (called group A), and neophytes (called group B). The data are:

Group	No. Raters	Est. No. Funct. Pts	Std. Dev./ Mean
Experienced	22	57.7	45.5
Group A	22	83.7	34.5
Group B	9	72.9	30.6

Low and Jeffery felt one organization's counts were exceptionally varying. The data excluding this organization are:

Group	No. Raters	Est. No. Funct. Pts	Std. Dev./ Mean
Experienced	18	51.5	21.6
Group A	11	83.7	41.2
Group B	9	72.9	42.0 [Low & Jeffery, 1990].

They also found large differences in reliability between organizations, but these differences could not be statistically analyzed.

The published account of Low and Jeffery's study implies that the raters were not provided a single set of counting rules. Thus, Low and Jeffery's study appears to look at the combined effect of inter-rater and inter-method reliability.

The following table summarizes the results of published sources about reliability:

Claims	Rater Experience	FP Methodology	Proj. Def.	Reliability
Dreger, 1989	Unknown	Unknown	Planned	Within 20%.
Dreger, 1989	Unknown	Unknown	Existing	Within 10%.
Rudolph, 1990	Unknown	Rel. 2.0	Fully Defined	Coef. of Var. within 10%.
Studies				
Rudolph, 1983	Unknown	Vague Rules	Requir. Doc	Within 30% of mean.
Low,Jeffery,'90	High	Various	Detailed Spec	Std. Dev. within 45.5% of mean.
Low,Jeffery,'90	Low	Various	Detailed Spec	Std. Dev. within 42% of mean.

In summary, the question of the inter-rater reliability of the Function Points metric is unresolved. Dreger and Rudolph claim that reliability is within 10% to 20% depending upon the level of application definition. In contrast, published studies indicate that for a planned application analyzed by experienced raters using differing counting rules, the standard deviation of the Function Point counts ranges from 30% to 45% of the mean count.

Determining the reliability of the metric is important because one of its uses is to measure the managerial impact of changes in tools or practices. The effects of these changes may be less than 30%, and may therefore be obscured by the inherent variability of the Function Point metric.

Inter-Method Reliability of the Function Points Metric

A number of variants to Albrecht's original Function Points methodology have been proposed. Some of these variants are summarized below. All of the variant metrics have been compared to the Albrecht Function Points metric is some fashion, but none of the articles which present the variants address the inter-method reliability issues.

In 1988, Symons published a number of criticisms of Albrecht's Function Points methodology and proposed his own version, called Mark II Function Points. Using data on 12 applications, Symons showed that Mark II Function Points were equivalent to Albrecht 1983 Function Points [Symons, 1988].

In 1989, Verner, et al. proposed a technology-dependent metric. Their approach is similar to Albrecht's Function Points methodology in that it uses the external functionality to estimate the size of a piece of software. It is different from Albrecht's Function Points in that its goal is to predict Source Lines Of Code (SLOC), and in that the components it looks at are classified differently than Function Types.

Verner, et al. took one application with 392 modules (functions or transactions), and defined four component types: file, menu, screen, and report. For each module, they counted the SLOC, the data elements referenced, relations referenced, choices, and report lines. They then developed equations for each component type using regression. The results are summarized in the following table. The numbers in the table represent the percentage of the modules for which the Albrecht, Mark II, and Verner et al. methodologies predicted the SLOC within 25% of the actual SLOC count.

	Albrecht	Mark II	Verner, et a.
Files	0%	-	68%
Menus	49%	0%	100%
Screens	42%	49%	80%
Reports	27%	24%	74%

Verner et al. concluded that their equations predicted module size better than the Albrecht Function Points methodology [Verner, et al., 1989].

In 1990, Ratcliff and Rollo adopted both the 1983 Albrecht Function Points methodology and Symons's Mark II methodology to Jackson System Development (JSD) [Ratcliff & Rollo, 1990]. Function Points analysis was developed in the context of traditional functional decomposition systems development methodologies. In contrast, JSD is an operational software development method. In the Jackson system, "life histories of entities are constructed, based on actions performed or experienced by those entities. ... [These] life histories model processes in the system. ... [These] processes communicate asynchronously with each other and the real world either by datastream or by state vector inspection".

Ratcliff and Rollo discussed some difficulties in adapting the two Function Point methodologies, but they succeeded in making counts of a JSD-based application. They reported that the adapted Albrecht and Mark II Function Point counts were within 0.4% of each other [Ratcliff & Rollo, 1990].

In summary, there are a number of variants of Albrecht's original Function Points methodology. As variants on the original methodology grow, this becomes a potential threat to reliability. Therefore, research is required to assess the degree to which inter-method reliability is a managerial concern.

3. RESEARCH QUESTIONS

Introduction

The study is divided into two sections. The first section addresses reliability. The second section is exploratory in nature and probes the factors that result in high or low reliability.

Inter-Rater and Inter-Method Reliability

As discussed in the Literature Review section, the reliability of the Function Points metric needs additional research. More specifically, there are two open questions, both of which are addressed by this study:

- What is the inter-rater reliability of the metric?
- What is its inter-method reliability?[4]

For each of these research questions, reliability is measured for each Function Type and for the unadjusted and adjusted Function Point counts.

This study looks at two sub-questions to the inter-method reliability issue. These are:

- How does the reliability different methodologies compare?
- How do counts made with one methodology compare with counts made with another?

The goal in studying the second sub-question is to identify a way to map counts prepared with one methodology onto counts prepared using another, and to measure the accuracy of this mapping.

Factors Which Affect Reliability

As was previously explained, this section is exploratory in nature and probes the factors that result in high or low reliability. The questions addressed are:

- How does experience affect reliability?
- How does the level of project definition affect reliability?

[4]Two methodologies are used here to study inter-method reliability.

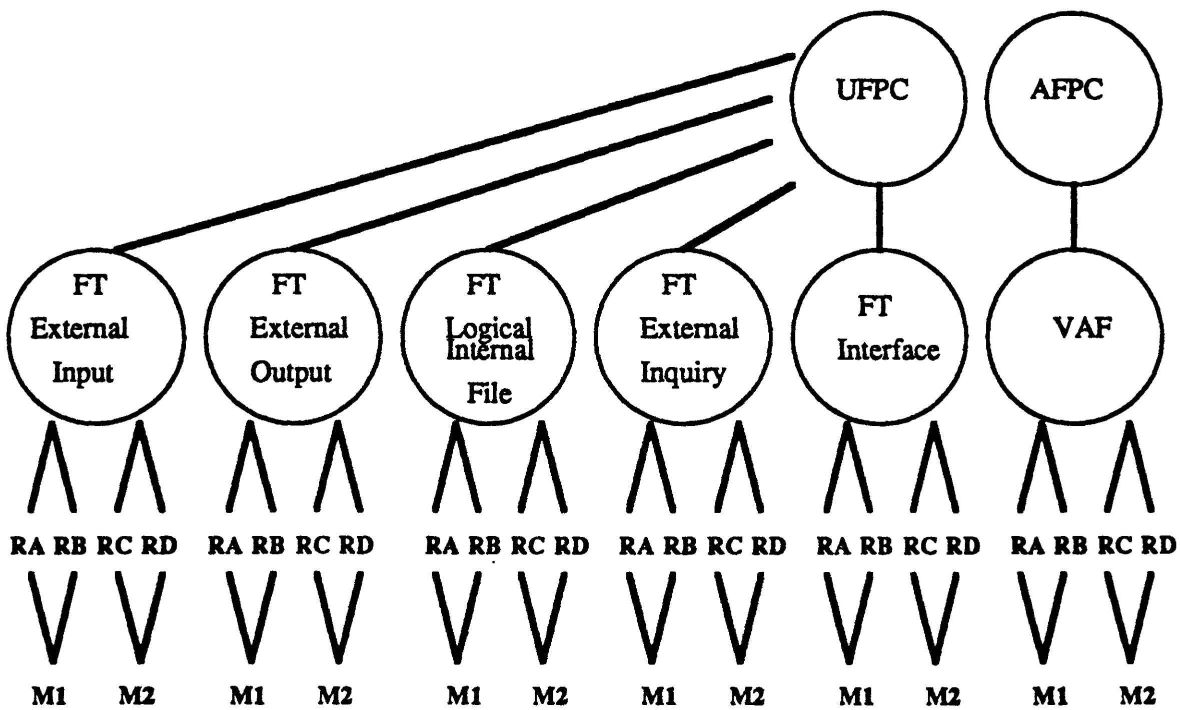
- How does familiarity with the application affect reliability?
- How does time spent on a count affect the application affect reliability?
- How does application size affect reliability?

4. RESEARCH METHODOLOGY

Introduction

The following illustrates the research model this study is employing. The model describes a Function Point count of one application development project. The abbreviations are defined below the diagram.

M1	Counting methodology 1 (IFPUG 3.0)
M2	Counting methodology 2 (E-R)
RA,RB,RC,RD	Different raters performing independent Function Point counts.
FT	Function Type Function Point counts
UFPC	Unadjusted Function Point Count
VAF	Value Adjustment Factor
AFPC	Adjusted Function Point Count



The overall approach for the study was to collect data in a field setting using two data collection instruments, and then to statistically analyze the data collected. The first data collection instrument was designed to capture data on actual applications. For each application, two pairs of Function Point counts were collected. Each pair was based on a different Function Points methodology. The second data collection instrument, which consisted of "micro-cases", was designed to explore previously identified issues.

Measures of Reliability

Reliability is measured using Spearman correlation coefficients and an average Magnitude of Relative Error (MRE) test. Correlation coefficients measure the degree of association between two variables. Since the data collected in this study are pairs of counts, correlation coefficients are an appropriate statistical measure of reliability. The average MRE test is also used because it provides an intuitive and more managerial measure of reliability. The average MRE test addresses the question, "on average, by what percentage will a particular Function Point count vary from a hypothetical average count?" In this study, MRE is defined as follows:

$$MRE = | (1 \text{ Rater's Count} - \text{Average Count}) / \text{Average Count} |$$

The advantage of an MRE test over a simple percent error test is that, because an absolute value is taken, positive and negative errors do not cancel each other when errors are averaged [Conte, et. al. 1986].

Comparison to the Low & Jeffery Study

Low and Jeffery provided a first look at reliability and identified some of the factors which affect it. This study is distinct from Low and Jeffery's in three important ways. First, the applications analyzed are believed to be more representative because they are actual applications rather than small examples. This study's use of actual applications provides a strong complement to Low and Jeffery's use of a case study. The second difference is that this study separates inter-rater reliability from inter-method reliability, and Low and Jeffery's study may not have.

The third difference is how reliability is measured. Low and Jeffery measured reliability as the coefficient of variation (standard deviation divided by the mean). Thus, when Low and Jeffery state that reliability is 33.8% to 45.5%, they mean that one standard deviation (or 68.26% if normally distributed) of counts will fall within this 33.8%-45.5% of the mean. This study uses average MRE, so its results represent the level within which 50% (or .675 standard deviations if normally distributed) of the counts will fall. Assuming that Function Point counts are normally distributed, Low and Jeffery's results can be converted to average MRE by multiplying by .675. This figure, designated the assumed Low and Jeffery average MRE, is 23%-31%.

Choice of Methodologies

Two Function Points methodologies are used in this study. The primary methodology is referred to as the IFPUG 3.0 methodology [Sprouls, 1990]. This methodology is documented in the IFPUG Function Point Counting Practices Manual Release 3.0. The methodology "is based on IBM CIS & A Guideline 313, dated November 1, 1984. The Function Point counting methodology described in 313 is commonly referred to as Albrecht 1984". The IFPUG 3.0 methodology is the latest version of the original Albrecht methodology, as clarified by GUIDE in 1982-1984 and IFPUG in 1986-1989. Since this methodology is the original and dominant methodology, it is an obvious choice of methodologies to include.

A second methodology is used in order to study inter-methodology reliability. This methodology is referred to as the Entity-Relationship Modeling (E-R) methodology. This methodology was originally developed by Jean-Marc Desharnais [Desharnais, 1988]. It has subsequently been modified by IFPUG, and will be included in a future release of the Function Point Counting Practices Manual [Porter, 1989]. The E-R methodology was chosen because it is likely to become the dominant methodology. One of the criticisms of the Albrecht methodology is that it uses the vocabulary and data constructs which were used in data processing in the mid 1970s. Now that relational databases and entity-relationship modeling have become the norm, there has been some demand for adapting the Function Points methodology. This is the reason an entity-relationship methodology is likely to become dominant. The IFPUG entity-relationship methodology in particular is likely to be dominant because IFPUG is the recognized Function Points standards-setting organization.

Study Activities

The research design, including the data collection instruments, was reviewed for methodological correctness by appropriate MIT Sloan School faculty, for participant acceptance by IFPUG counting practices committee members, and for clarity and accuracy by prominent members of IFPUG. Appendix A contains a copy of the cover letter which requested comments on the research design.

Letters were mailed to all IFPUG members soliciting their participation in the study. The members were asked to commit to the study and identify coordinators at each participating site. Each participating site was assigned a site number. Appendix B contains a copy of the letter soliciting participation.

The packets sent to the sites included the following items. A copy of each of these items is included in an appendix, noted below:

- A cover letter (Appendix C)
- Instructions for the coordinator (Appendix D)
- Two copies each of two counting manuals: one based on the IFPUG 3.0 methodology (Appendix E), and the other based on the E-R methodology (Appendix F).
- Four forms entitled "Form for Rater" (One sample in Appendix H)
- A form entitled "Questionnaire for Coordinator" (Appendix I)

Five pre-addressed and pre-stamped return envelopes were also included in each packet to allow raters to independently return their results.

For the first data collection instrument, the coordinator for each site selected one or two in-house applications; and, for each application, was requested to choose up to four people to make independent Function Point counts. For each application, two of the counts were made using the IFPUG 3.0 methodology (designated Raters A & B), and the other two were made using the E-R methodology (designated Raters C & D). The raters recorded the results of their counts, a description of their experience, and some information about the count on questionnaires. Each rater mailed his questionnaire directly to the researchers.

For the second data collection instrument, each site's coordinator completed a questionnaire. The questionnaire inquired about the type of work the site does, what it uses Function Points for, and the counting conventions it employs. The questionnaire included a number of "micro-cases" designed to detail the site's practices regarding certain possibly contentious areas. A copy of the Questionnaire for Coordinator is contained in Appendix I.

Data analysis was done using the SAS Institute's Statistical Analysis System.

5. RESULTS OF INTER-RATER & INTER-METHOD STUDY

Background Data

The quantities of data used in this study are summarized below:

Measure	Number
Potential Sites	63
Actual Sites	25
Raters	120
Inter-Rater Pairs, IFPUG 3.0 Method	29
Inter-Rater Pairs, E-R Methodology	23
IFPUG 3.0 Applications	37
E-R Applications	31
Inter-Method Pairs	28

Potential sites are those who consented to participate in the study in response to a letter which was sent to all IFPUG members. Actual sites are those who provided at least one pair of counts. Inter-rater pairs refers to the number of applications for which there are a pair of counts based on the same methodology. IFPUG 3.0 Applications and E-R Applications refer to the number of applications for which there are at least one count (some sites sent only one count for one of the methodologies). Inter-method pairs refers to the number of applications for which there is at least one count based on the IFPUG 3.0 methodology and one count based on the E-R methodology. When two counts per methodology were available, the inter-method pair consisted of: 1) the average of the IFPUG 3.0 pair and 2) the average of the E-R pair.

Appendix G lists the participating sites.

The overall response rate was 40% This rate is quite high given the amount of effort required of each site. The average counting time per site is 21 hours, which amounts to over 500 work-hours of counting time. This high participation rate was made possible by follow up calls and letters.

The following table describes the study participants and the counts.

	Average	Range
Participants		
IS Experience (yrs)	10.9	0-28
FP Analysis Experience (yrs)	1.3	0-5
Formal FP Training (days)	1.5	0.5-5
Counts		
Count Time (hrs)	4.6	0.5-18
Number of Function Points	582	70-3613

The majority of the participants worked in Manufacturing or Finance firms. Appendix M provides a detailed breakdown of the types of firms which participated.

The majority of the applications included are batch or interactive MIS applications in the accounting and finance area. Appendix K provides a detailed breakdown of the types of applications which comprise the study.

Inter-Rater Reliability

The following table lists the correlation coefficients and the average Mean Relative Errors (MRE) for the first three levels of Function Points analysis. The table includes data from all raters who used the IFPUG 3.0 methodology. The purpose of this table is to quantify the reliability of Function Points analysis and to illuminate which areas of the Function Points methodology are the most reliable and which are the least.

Function Type	Spearman Correlation Coefficients (Significance Levels)[5]				Avg. MRE
	Component Counts (CC)			FTFPC	FTFPC
	Low	Avg	High		
Logical Int. Files	.83(.000)	.86(.000)	.13(.490)	.73(.000)	18.6%
External Interfaces	.75(.000)	.15(.456)	.16(.403)	.61(.001)	47.5%
External Inputs	.50(.006)	.56(.002)	.64(.000)	.74(.000)	18.2%
External Outputs	.81(.000)	.53(.003)	.73(.000)	.87(.000)	12.4%
External Inquiry	.77(.000)	.24(.202)	.63(.000)	.70(.000)	35.0%
Unadjusted Function Point Count (UFPC)				.91(.000)	10.3%

All of the above correlation coefficients were based on up to 29 pairs of Function Points counts. However, since not all applications include components having all Function Types at all complexities, some of the data pairs consist of two zeros. The following table provides the number of pairs of non-zero data points for each correlation coefficient.

	Number of Pairs of Non-Zero Data Points			
	Component Counts			FTFPC
Function Types	Low	Avg	High	
Logical Internal Files	27	19	14	29
External Interfaces	22	17	9	24
External Inputs	27	27	19	29
External Outputs	24	29	25	29
External Inquiry	19	20	13	24
				# Pairs
Unadjusted Function Point Count				29

The two most striking conclusions which can be drawn from this data are that 1) the reliability is quite high and 2) that there is a general improvement in reliability as one moves from the Component Count to the Function Type Function Point Count to the Unadjusted Function Point Count. These conclusions will be discussed in more detail later.

The following table summarizes the General Systems Characteristics (GSCs) and the correlation coefficients and Mean Relative Errors for unadjusted and adjusted Function Point counts. As with the previous table, this one includes data from all raters who

[5]Significance Level: probability this result could have achieved by chance

used the IFPUG 3.0 methodology. Its purpose is to quantify the reliability of Function Points analysis and to illuminate which areas of the Function Points methodology are the most reliable and which are the least.

General Systems Characteristic	Spearman Correlation Coefficient (Significance Level)
Data Communication	.67(.000)
Distributed Function	.43(.022)
Performance	.70(.000)
Heavily Used Configuration	.64(.000)
Transaction Rates	.48(.010)
On-Line Data Entry	.74(.000)
Design for End User Efficiency	.73(.000)
On-Line Update	.70(.000)
Complex Processing	.48(.001)
Usable in Other Applications	.62(.001)
Installation Ease	.56(.002)
Operational Ease	.49(.009)
Multiple Sites	.49(.009)
Facilitate Change	.51(.006)

	Spearman Correlation Coefficient (Significance Level)	Avg. Mean Relative Error
Unadjusted Function Point Count (UFPC)	.91(.000)	10.3%
Sum of General Systems Characteristics	.58(.001)	10.6%
Adjusted Function Point Count (AFPC)	.93(.000)	10.5%

All of the above correlation coefficients were based on 29 pairs of Function Points counts, all of which were non-zero.

From the GSC and AFPC data, one can conclude that the reliability of the GSCs is lower than that of the UFPC, but clearly significant at the 99.9% confidence level. Also, one can conclude that the GSCs have relatively little impact on the reliability of the AFPC.

Based on all the inter-rater data presented, the following conclusions may be drawn:

1. The FTFPC for a given Function Type is more reliable than the CCs for that Function Type. For example, the FTFPC for External Inputs is .737, while

CCs for Low, Average, and High complexity External Inputs are .494, .559, and .641. While an individual CC may have a higher correlation coefficient than the FTFPC, in all cases the FTFPC's correlation coefficient is higher than a simple average of the correlation coefficients for the CCs. This may indicate that, for a given component, raters have difficulty determining the relative complexity. Since this difficulty affects where the component gets counted and not whether the component gets counted, it has a much greater affect on the CCs than on the FTFPC. In other words, the differences which reduce the reliability of the CCs appears to offset each other in the FTFPC.

2. Function Point counts of External Inputs and External Outputs have the highest reliability, and those of External Interfaces and External Inquiries have the lowest reliability. However, the low reliability of counts of External Interfaces may be partially due to there being too little non-zero data.

3. The UFPC is more reliable than any of the five FTFPCs. The explanation is identical to that of conclusion 1. The difference in reliability may indicate that, for a given component, raters have difficulty determining the Function Type. Since this difficulty affects where the component gets counted and not whether the component gets counted, it has a much greater affect on the FTFPCs than on the UFPC. In other words, the differences which reduce the reliability of the FTFPCs offset each other in the UFPC.

4. In contrast to conclusions 1 and 3, the reliability of the sum of the General Systems Characters (GSCs) is no higher than the reliabilities of the individual GSCs. This is because each GSC is independent (i.e. assigning a value to a GSC is not a matter of deciding where something should be counted), and no systematic offsetting of differences occurs.

5. The reliability of the General Systems Characteristics is less than the UFPC's or AFPC's reliability, although it never approaches the level where one could conclude that any correlation is due to happenstance.

6. The reliability of the bottom line, (i.e. the AFPC) is quite high. The correlation coefficient is .93, and the average Mean Relative Error (MRE) is 10.5%. These figures are in line with Rudolph's and Dreger's recent observations and indicate that the Function Points metric is more reliable than Low and Jeffery's data suggests. The difference between this study's results and Low and Jeffery's is believed to be attributable to this study's use of uniform counting rules. A second possible explanation is that the small size of Low

and Jeffery's applications may have caused lower reliability. Counts of smaller applications may have lower reliability because there is less opportunity for differences to offset each other. The affect of application size is discussed in the Results of Exploratory Study section.

Inter-Method Reliability

As was discussed in the Research Questions section, there are two sub-questions to Inter-Method reliability. These are:

1. How does the reliability of counts made with the IFPUG 3.0 methodology compare the reliability of with those made with the E-R methodology?
2. How do counts made with the IFPUG 3.0 methodology compare with counts made with E-R methodology?

To answer the first sub-question, the correlation coefficients and MREs of counts made with the two methodologies are compared. The following table provides the results of the counts made with the Entity-Relationship (E-R) methodology.

Function Type	Spearman Correlation Coefficients (Significance Levels)				Avg. MRE
	Component Counts (CC)			FTFPC	FTFPC
	Low	Avg	High		
Logical Int. Files	.73(.000)	.48(.020)	.28(.188)	.71(.000)	20.8%
External Interfaces	.79(.000)	-.25(.253)	.64(.001)	.72(.001)	37.8%
External Inputs	.67(.001)	.58(.003)	.63(.001)	.94(.000)	22.8%
External Outputs	.42(.046)	.41(.055)	.62(.002)	.73(.000)	28.3%
External Inquiry	.58(.004)	.82(.001)	.89(.000)	.86(.000)	36.2%
Unadjusted Function Point Count (UFPC)				.75(.000)	16.6%

All of the above correlation coefficients were based on 23 pairs of Function Point counts. The following table lists the number of pairs of non-zero data for each correlation coefficient.

	Number of Pairs of Non Zero Data Points			
	Component Counts			FTFPC
Function Types	Low	Avg	High	
Logical Internal Files	22	20	7	23
External Interfaces	15	12	3	17
External Inputs	21	21	16	23
External Outputs	20	21	20	23
External Inquiry	17	16	11	20

	# Pairs
Unadjusted Function Point Count	23

From this data, one can conclude that 1) the reliability of counts made with the E-R methodology are not as high as the reliability of counts made with the IFPUG 3.0 methodology, 2) that there is a general improvement in reliability as one moves from the Component Count to the Function Type Function Point Count (FTFPC), but not as one moves from the FTFPC to the Unadjusted Function Point Count. These conclusions will be discussed in more detail later.

The following table summarizes the Correlation Coefficients for the General Systems Characteristics and the correlation coefficients and average Mean Relative Errors for unadjusted and adjusted Function Point counts. As with the previous table, this one includes data from all raters who used the E-R methodology.

General Systems Characteristic	Spearman Correlation Coefficient (Significance Level)
Data Communication	.30(.157)
Distributed Function	.51(.014)
Performance	.28(.190)
Heavily Used Configuration	.30(.160)
Transaction Rates	.11(.631)
On-Line Data Entry	.34(.114)
Design for End User Efficiency	.42(.045)
On-Line Update	.55(.007)
Complex Processing	.32(.141)
Usable in Other Applications	.41(.054)
Installation Ease	.62(.002)
Operational Ease	.18(.422)
Multiple Sites	.66(.001)
Facilitate Change	.69(.000)

	Spearman Correlation Coefficient (Significance Level)	Avg. Mean Relative Error
Unadjusted Function Point Count (UFPC)	.75(.000)	16.6%
Sum of General Systems Characteristics	.56(.006)	15.9%
Adjusted Function Point Count (AFPC)	.78(.000)	17.3%

All of the above correlation coefficients were based on 23 pairs of Function Points counts, all of which were non-zero.

From the General Systems Characteristics and Adjusted Function Point Count data, one can conclude that the reliability of the GSCs developed from the E-R methodology are lower than the reliabilities of the UFPC and AFPC.

The following table summarize the comparison of the reliability of the IFPUG 3.0 and the E-R methodologies.

	IFPUG 3.0		E-R	
	Corr.Coef.	MRE	Corr.Coef.	MRE
Function Types				
Logical Internal Files	.73	18.6%	.71	20.8%
External Interfaces	.61	47.5%	.72	37.8%
External Inputs	.74	18.2%	.94	22.8%
External Outputs	.87	12.4%	.73	28.3%
External Inquiry	.70	35.0%	.86	36.2%
Unadjusted Function Point Count	.91	10.3%	.75	16.6%
Sum of GSCs	.58	10.6%	.56	15.9%
Adjusted Function Point Count	.93	10.5%	.78	17.3%

From these three tables, one can determine:

1. Looking only at the bottom line (i.e. the AFPC), counts made with the IFPUG 3.0 methodology have higher reliability than those made with the E-R methodology. While it is impossible to determine exactly why the IFPUG 3.0 methodology is more reliable, one possible explanation is that the raters who used the IFPUG 3.0 methodology had a higher experience level than those who used the E-R methodology. A look at the raters' experience levels disproves this hypothesis. The average experience levels are:

	Applications Development Experience	Function Points Experience
Raters using IFPUG 3.0	10.6 yrs	1.4 yrs
Raters using E-R	11.2 yrs	1.7 yrs

A more plausible explanation is that the difference in reliability is caused by differences in the raters' experience levels with the methodologies. All raters were provided counting manuals and instructed to use these manuals exclusively, so all raters used a methodology which was new to them. However, the IFPUG 3.0 methodology is quite similar to the methodologies the raters currently work with, and the E-R methodology is entirely new. Consequently, the raters who used the IFPUG 3.0 methodology may have had an advantage.

The differences in familiarity with the counting manuals were exacerbated by the fact that Entity-Relationship data modelling was new to some raters who used the E-R methodology. Half of these raters had used E-R modeling, one quarter had not used E-R modeling but had some training in it, and one quarter had neither used nor studied it.

A third potential hypothesis for the inter-method differences in reliability is that there are differences in quality (i.e. readability, clarity of the documentation) of the counting manuals.

2. Counts made with the E-R methodology do not display the offsetting of differences that counts made with the IFPUG 3.0 methodology display. Specifically, based on correlation coefficients, the UFPC is no more reliable than FTFPCs.

   A possible explanation is that the E-R methodology inherently *does* produce the same offsetting of differences, but this effect is being overwhelmed by errors or omissions made by the raters. These errors and omissions may be caused by the raters' lack of familiarity with the E-R methodology. In other words, because they are not familiar with the methodology, raters are not only having difficulty deciding where a component gets counted, they may be also omitting components and making other errors.

3. The E-R methodology counts confirm two of the conclusions drawn from looking at the IFPUG 3.0 counts:

   - Each FTFPC is more reliable than the CCs for that Function Type.
   - The External Interfaces and External Inquiry Function types are the most problematic.
   - The reliability of the General Systems Characteristics are relatively low. Note that both methodologies use exactly the same GSC rules and calculations.

The second inter-method sub-question, 'How do counts made with the IFPUG 3.0 methodology compare with counts made with E-R methodology?', was answered via regression analyses. Average IFPUG 3.0 count was the dependent variable and average E-R count was the independent variable. The intercept was set to zero. If the two methodologies produced identical results, the slope of the regression line would be one, and the R-Squared measure of fit would be 1.00.

For a given application, an average of rater A's and rater B's UFPC is used as the actual dependent variable value, and an average of rater C's and rater D's UFPC is used as the independent variable value. When only one count per methodology is available, the one available UFPC is used instead of an average UFPC.

The following table presents the results of the regression analyses.

	# Data Points	Slope	R-Squared
Function Types			
Logical Internal Files	28	.936	.715
External Interfaces	18	.867	.619
External Inputs	28	.957	.896
External Outputs	28	.992	.957
External Inquiry	28	.849	.878
Unadjusted Function Point Count	28	.971	.925

Note that the slope is the number one would multiply an E-R count by to get an approximately equivalent IFPUG 3.0 count.

Figures 5.1 through 5.6 provide plots of the relationship between the IFPUG 3.0 and E-R FTFPCs and UFPC.

Figure 5.1
IFPUG v. E-R Method for Logical Internal Files

PLOT OF AVAB_FIL*AVCD_FIL   SYMBOL USED IS +

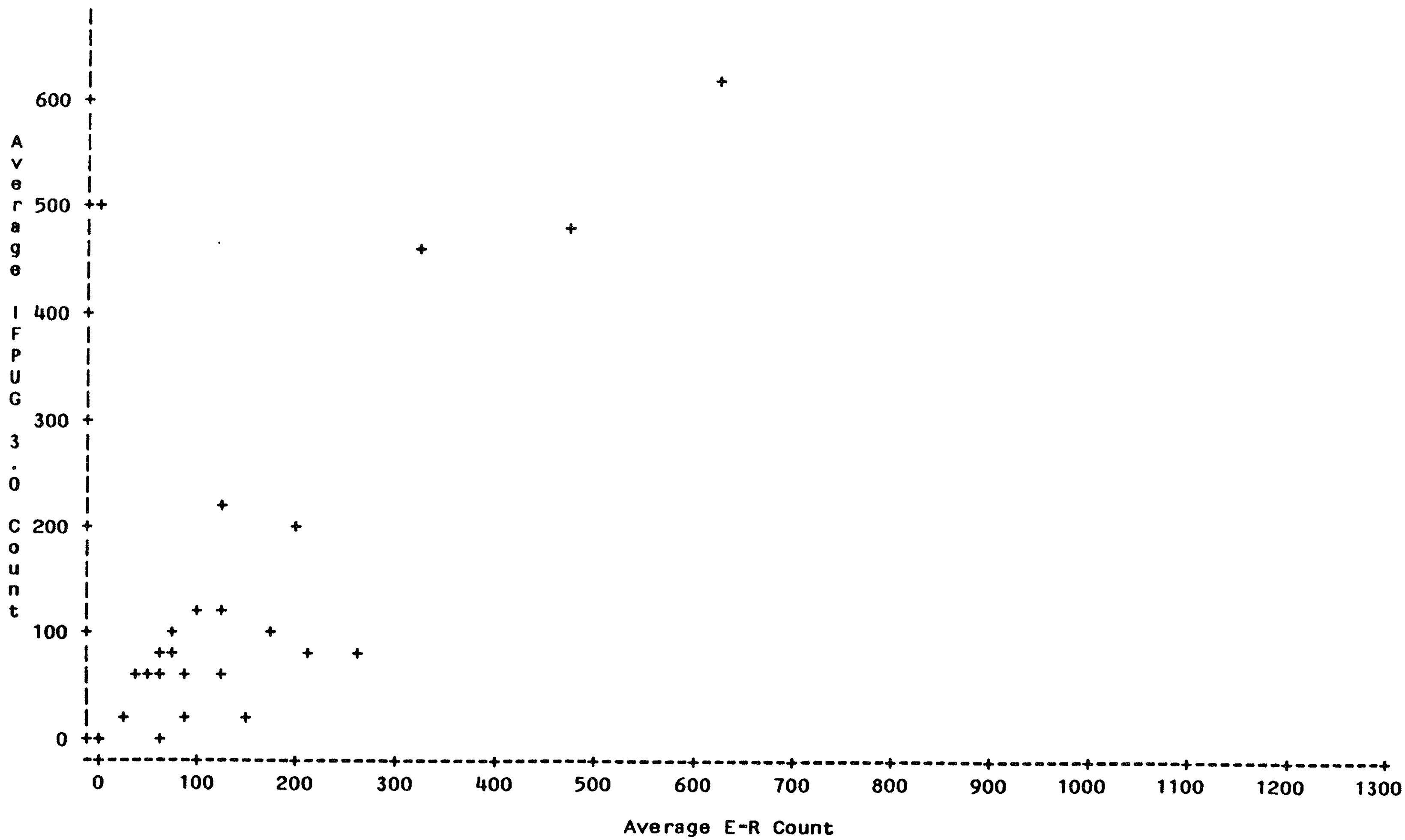


Figure 5.2
IFPUG v. E-R Method for External Interfaces

PLOT OF AVAB_INT*AVCD_INT   SYMBOL USED IS +

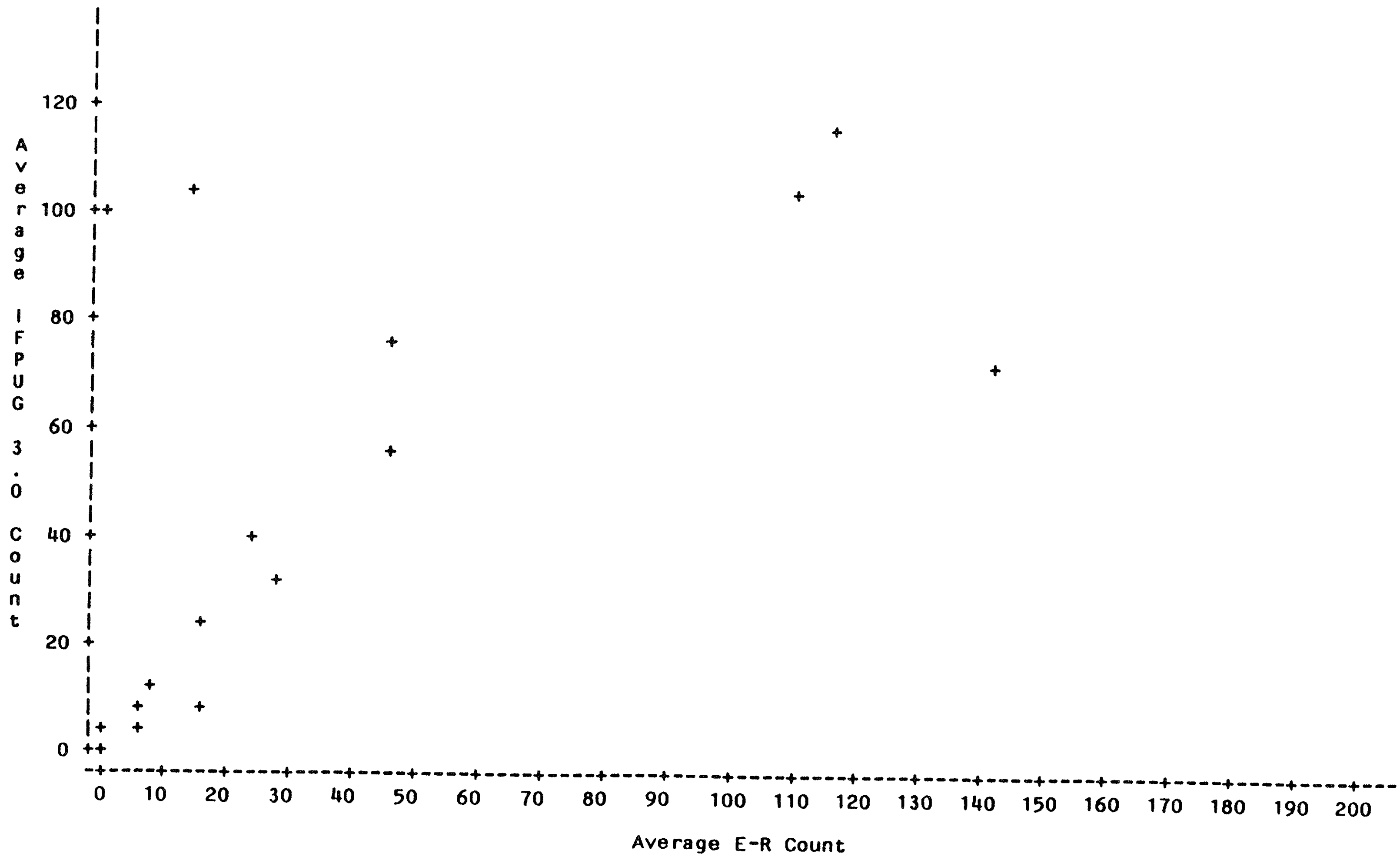


Figure 5.3
IFPUG v. E-R Method for External Inputs

PLOT OF AVAB_INP*AVCD_INP   SYMBOL USED IS +

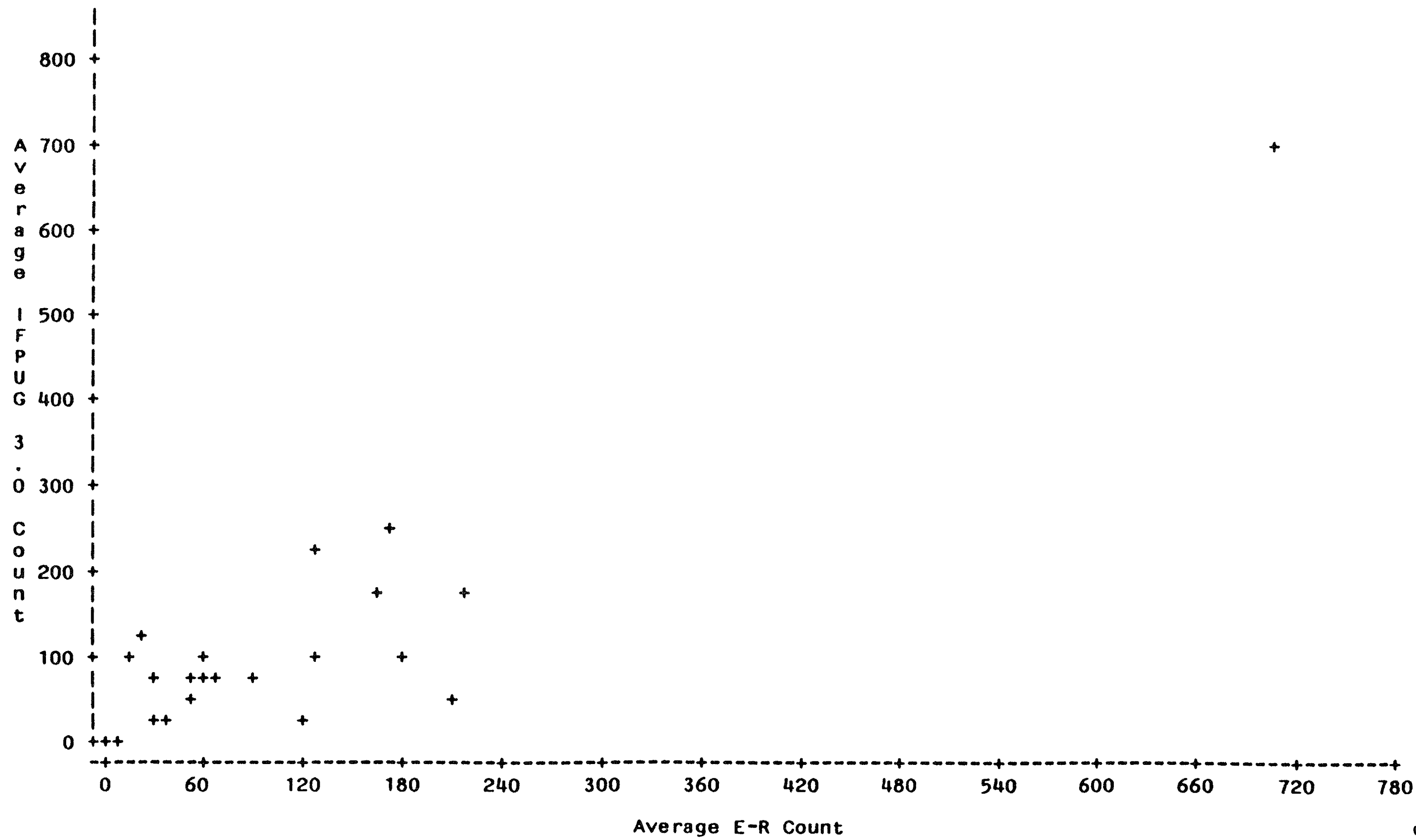


Figure 5.4
IFPUG v. E-R Method for External Outputs

PLOT OF AVAB_OUT*AVCD_OUT   SYMBOL USED IS +

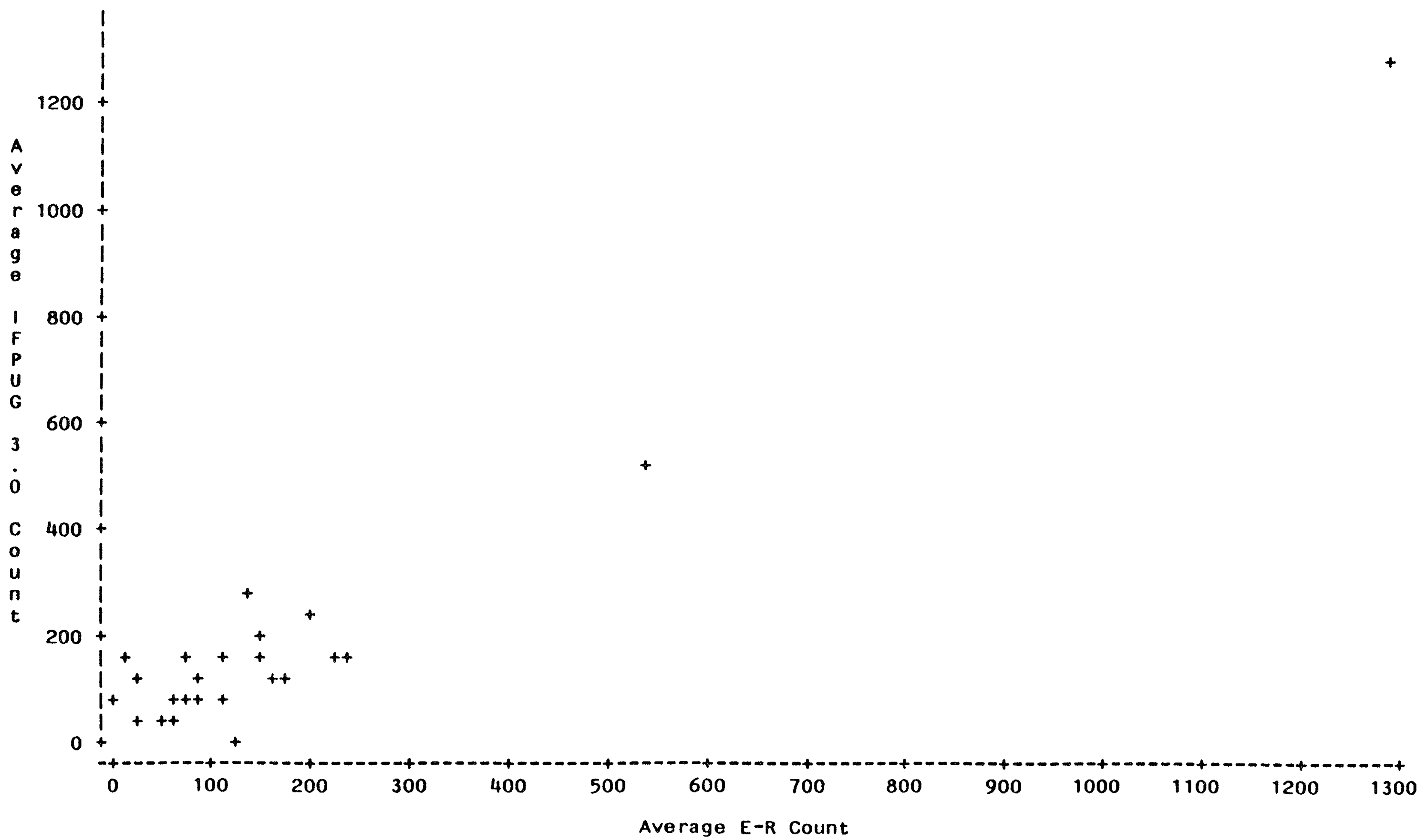


Figure 5.5
IFPUG v. E-R Method for Inquiries

PLOT OF AVAB_INQ*AVCD_INQ   SYMBOL USED IS +

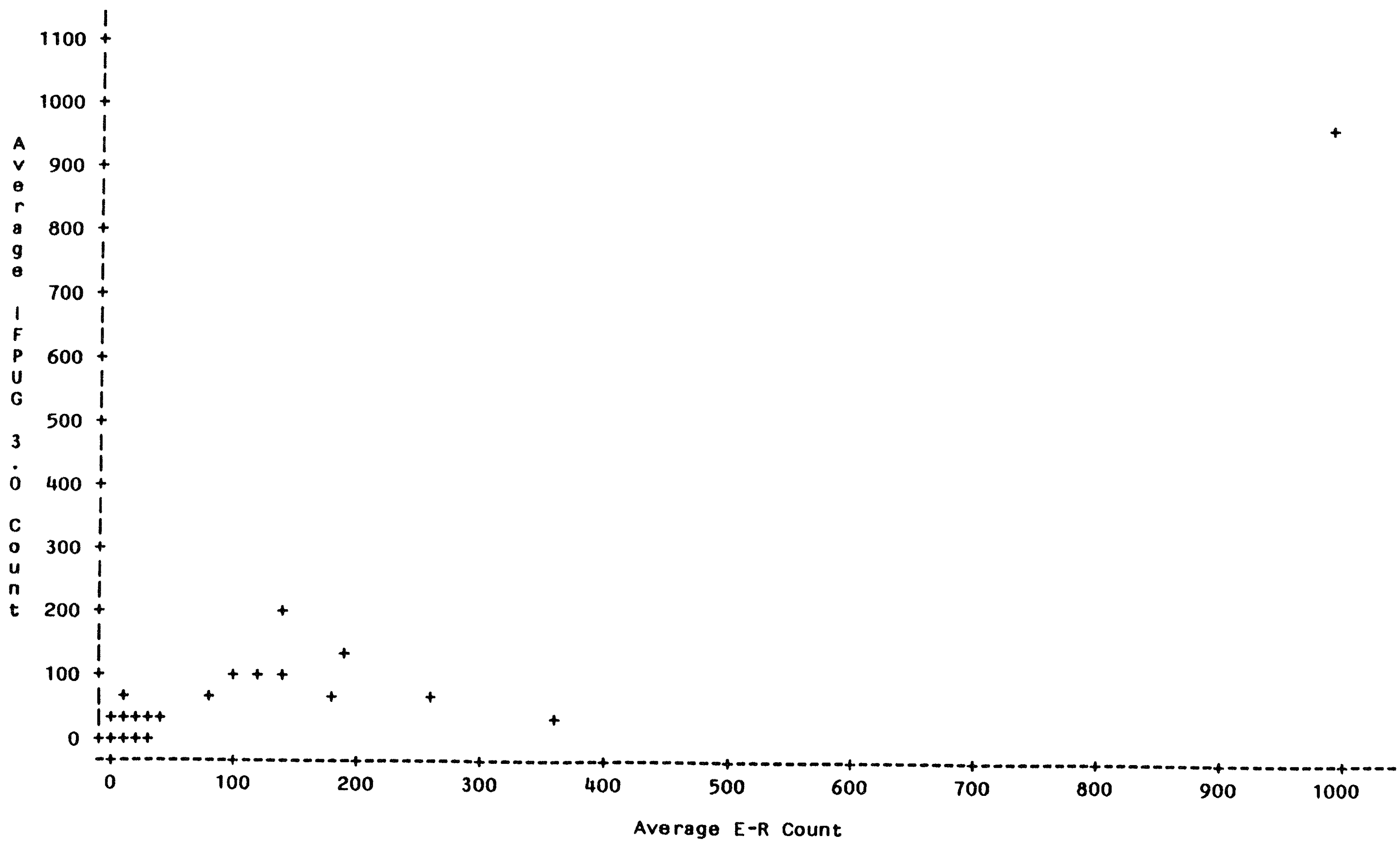
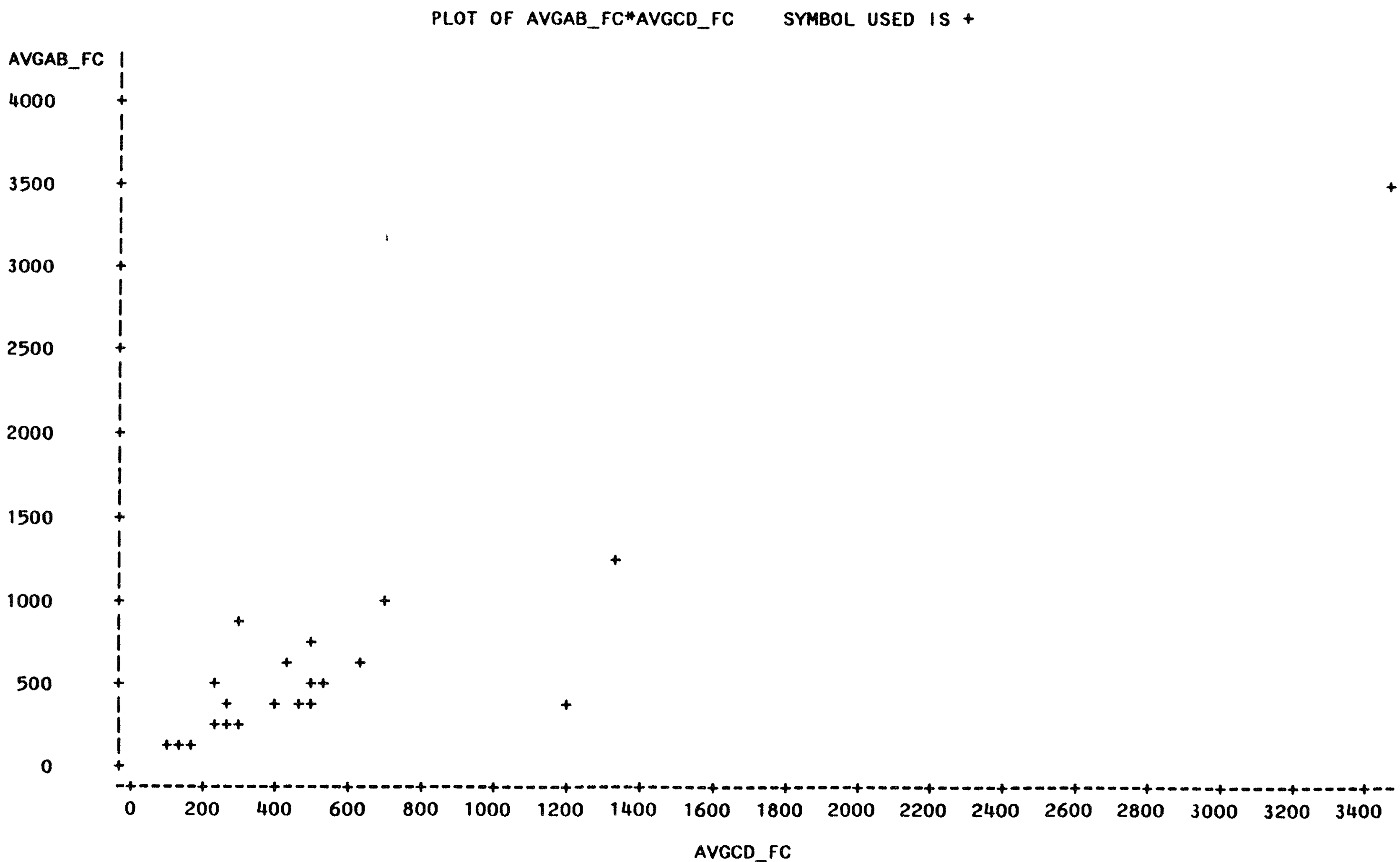


Figure 5.6
IFPUG v. E-R Method for Undajusted Function Pt Counts



As can be seen from the plots, the data include one application which is much larger than the others. The counts for this application are also very highly correlated. Because of this application's large size and high correlation, it has a large influence over the slope and R^2 values of the regression analyses. To ensure that this application is not inappropriately influencing the analysis, the regressions were run without it. The following table summarizes the results:

	# Data Points	Slope	R-Squared
Function Types			
Logical Internal Files	27	.914	.652
External Interfaces	18	.867	.619
External Inputs	27	.873	.719
External Outputs	27	.973	.858
External Inquiry	27	.416	.562
Unadjusted Function Point Count	27	.916	.803

Comparing the results of the two sets of regressions (including and not including the outlying application), it is evident that the outlying application does have an influence. The slope of the regression has changed from .971 to .916. The majority of the difference appears to lie in the External Inquiry Function Type, the slope of whose regression line has changed from .849 to .416.

From these regression analyses, one can conclude the following:

- The difference between the UFPCs developed using the IFPUG 3.0 methodology and those developed using the E-R methodology are minor. For this dataset, UFPCs made using the E-R methodology can be converted into equivalent IFPUG 3.0 methodology UFPCs by multiplying by a number between 0.92 and 0.97.

- With the exception of the External Inquiry Function Type, the differences between the FTFPC's developed using the IFPUG 3.0 methodology and the E-R methodology are also minor. For this dataset, the differences in External Inquiry FTFPCs are significant. The IFPUG 3.0 method results in Function Point counts that are between 42% and 85% of the counts produced from the E-R methodology.

- Based on the high R^2 values, there is good agreement between FTFPCs and UFPCs developed using the two methodologies.

- This comparison, coupled with the previously discussed results of Symons, Verner et al, and Ratcliff & Rollo; indicates that the concept of Function Points analysis is very robust. That is, the data indicates that counts made with a variety of Function Points methodologies are highly correlated.

6. RESULTS OF EXPLORATORY STUDY

General

As was discussed in the Research Questions section, this study explores the factors affecting reliability by addressing the following questions:

- How does experience affect reliability?
- How does the level of project definition affect reliability?
- How does familiarity with the application affect reliability?
- How does time spent on a count affect the application affect reliability?
- How does application size affect reliability?

Each of these questions will be studied independently. For each question, the data have been divided into appropriate groups, and the reliability of each group is measured. Only the reliability of the Adjusted Function Point Count (AFPC) is presented. As in the previous sections, the reliability measures used are Spearman correlation coefficients and average Mean Relative Error (MRE).

Since the data consist of pairs of counts, a group consists of all pairs of counts where both raters have the same characteristic. For example, the "Familiar with Application" group consists of all pairs of counts where both raters have worked on the application. For each study, there is a group which consists of pairs of counts by raters who do not fit any other group because the raters do not share a characteristic. For example, the "Mixed Familiarity" group consists of all pairs of counts where one rater has worked on the application and the other rater has not.

All of the studies are based on a dataset which merges the counts made with the IFPUG 3.0 and E-R methodologies. The data were merged in order to increase the size of the data pool and enable more types of studies to be conducted. Given the high correlation observed in the previous section, this is believed to be appropriate. There are up to 51 pairs of counts for each study. The correlation coefficient of the AFPC for this merged dataset is 0.85 with a significance level of 0.000. The MRE is 13.6%.

Affect of Experience on Reliability

This study uses Low and Jeffery's experience topology [Low & Jeffery, 1990]. The counts are divided into three groups (plus the Mixed Experience group) depending upon the raters' years of experience with applications development and Function Points analysis. The groups are defined as follows:

	Applications Development	Function Points
Neophytes	< 2 yrs	< 2yrs
Moderately Experienced	>= 2 yrs	< 2 yrs
Experienced	>= 2 yrs	>= 2 yrs

The results of the analyses are summarized in the following table:

	Number of Pairs	Correlation Coefficient	Avg. MRE
Neophytes	2	-	12.4%
Moderately Experienced	10	.85(.002)	7.7%
Experienced	11	.86(.001)	20.9%
Mixed Experience	22	.86(.000)	10.0%
Entire Dataset	51	.85(.000)	13.6%

There are sufficient data to make only the most tenuous of conclusions. Since none of the correlation coefficients is significantly different than the correlation coefficient for the entire dataset, the available data suggest that experience level does not significantly affect Function Points reliability.

This conclusion agrees with the Low and Jeffery data which include their outlier point, but disagrees with their conclusion.

Affect of Project Definition on Reliability

The data have divided into four groups. The Requirements Definition group consists of pairs of counts based on a requirements analysis document (or a requirements analysis document and an incomplete external design). The External Design group consists of pairs of counts based on external design documents such as hardcopy of example screens, reports, file layouts, etc (or design documents and an implemented system). The Implemented System group consists of pairs of counts based on the actual, implemented system. The Mixed Basis group consists of counts where each of the raters used a different basis.

Function Points analysis measures "the function value delivered to the user, as viewed through a software program's external attributes" [Albrecht, 1979]. Since functionality is established during the requirements definition phase of project development, Function Point counts for the same application should not vary with the basis of the count.

The results of the analyses are summarized in the following table:

	Number of Pairs	Correlation Coefficient	Avg. MRE
Requirements Definition	4	.80(.200)	20.9%
External Design	20	.95(.000)	6.4%
Implemented System	3	.50(.667)	15.1%
Mixed Basis	18	.69(.001)	18.9%
Entire Dataset	51	.85(.000)	13.6%

There is insufficient variety of data to draw any conclusions.

Affect of Familiarity with Application on Reliability

The data have been divided into three groups. The Familiar group consists of pairs of counts where both raters have worked on the application. The Not Familiar group consists of pairs of counts where both raters have not worked on the application. The Mixed Familiarity group consists of pairs of counts where one rater is familiar with the application, and the other is not.

	Number of Pairs	Correlation Coefficient	Avg. MRE
Familiar with Application	2	-	9.3%
Not Familiar w. Applic.	35	.88(.000)	11.7%
Mixed Familiarity	14	.70(.000)	19.0%
Entire Dataset	51	.85(.000)	13.6%

Again, there is insufficient variety of data to draw any conclusions.

Affect of Counting Time on Reliability

To analyze the affects of time spent on the Function Point count, each rater's counting rate was calculated (Function Points per hour). One can hypothesize that faster counting rates will show lower reliability, simply because the slower raters are being more meticulous and making less errors. This hypothesis assumes that counting rates are driving reliability rather than just being symptomatic of some experience variable which is actually driving reliability.

The data were grouped by counting rates using two separate splits. With the first split, the data were divided into two approximately equal groups. The results of this split are summarized in the following table.

	Number of Pairs	Correlation Coefficient	Avg. MRE
Counting Rate < 70 FP/Hr	17	.89(.000)	12.6%
Counting Rate > 70 FP/Hr	21	.75(.000)	12.4%
Mixed Counting Rates	13	.77(.002)	16.9%
Entire Dataset	51	.85(.000)	13.6%

In the second split, the data were divide into three equal groups. The results of this split are summarized below.

	Number of Pairs	Correlation Coefficient	Avg. MRE
Counting Rate < 60 FP/Hr	11	.90(.000)	7.4%
60 < Counting Rate < 150	4	-	23.6%
Counting Rate > 150 FP/Hr	13	.86(.000)	7.0%
Mixed Counting Rates	12	.88(.000)	12.3%
Entire Dataset	51	.85(.000)	13.6%

The available data do not show any consistent pattern which relates counting rate to reliability.

Affect of Application Size on Reliability

One can hypothesize that small applications will show lower reliability, because on small applications there is less opportunity to get the effects of the offsetting of differences than there is on larger projects.

To analyze the affect of application size on reliability, the data were grouped by average Adjusted Function Point Count (APFC) using two separate splits. With the first split, the data were divided into two approximately equal groups. The results of this split are summarized in the following table.

	Number of Pairs	Correlation Coefficient	Avg. MRE
Applications with avg. AFPC < 320	23	.50(.014)	12.8%
Applications with avg. AFPC > 320	28	.58(.001)	14.2%
Entire Dataset	51	.85(.000)	13.6%

In the second split, the data were divide into three equal groups. The results of this split are summarized below.

	Number of Pairs	Correlation Coefficient	Avg. MRE
Applications with avg. AFPC < 225	16	.63(.001)	8.8%
Applic's with 225 < avg. AFPC < 475	17	.31(.224)	15.6%
Applications with avg. AFPC > 475	18	.50(.035)	15.9%
Entire Dataset	51	.85(.000)	13.6%

The available data do not show any consistent pattern which relates application size to reliability. Therefore, the hypothesis that small applications have lower reliability is refuted.

7. RESULTS OF SENSITIVITY ANALYSES

The purpose of this section is to test the robustness of the inter-rater and inter-method reliability results. Two potentially confounding hypotheses are addressed:

- The inter-rater results are inappropriately influenced by the outlier application.

- The inter-method results are due to factors such as differences in the raters' experience levels, or the amount of time spent on the counts, rather than inherent differences in the methodologies.

The outlier referred to in the first hypothesis is an application which has a much larger Function Point Count than the other applications. This application's counts are highly correlated. Thus, it is possible that this point has an inappropriate level of influence over the measures of reliability. This outlier was discussed in the Inter-Method Results section.

To address the question of the influence of the outlier, all of the statistics were run without the outlier application. The following table compares the results:

	With Outlier		Without Outlier	
	Corr.Coef.	MRE	Corr Coef.	MRE
Function Types				
Logical Internal Files	.73(.000)	18.6%	.70(.000)	19.3%
External Interfaces	.61(.001)	47.5%	.59(.001)	45.2%
External Inputs	.74(.000)	18.2%	.71(.000)	18.9%
External Outputs	.87(.000)	12.4%	.85(.000)	12.9%
External Inquiry	.70(.000)	35.0%	.67(.000)	36.5%
Unadjusted Function Point Count	.91(.000)	10.3%	.90(.000)	16.6%
Sum of GSCs	.58(.001)	10.6%	.58(.000)	10.6%
Adjusted Function Point Count	.93(.000)	10.5%	.92(.000)	10.8%

This data demonstrates that the outlier point does have an influence. For the Function Type Function Point Counts (FTFPCs), removing the outlier generally consistently reduces the correlation coefficients by .02 - .03 and generally the MRE by about 1%. However, at the Unadjusted Function Point Count (UFPC) and Adjusted Function Point Count (AFPC) levels, the affect of removing the outlier is negligible.

In conclusion, removing the outlier has little impact on the measures of reliability, and certainly no impact on the conclusions presented in this study.

Regarding the second hypothesis, that the inter-rater results do not really measure differences in methodology, the Results of Exploratory Study section has already demonstrated that experience, project definition, familiarity with application, and application size do not appear to affect reliability. Consequently, differences in these factors cannot influence the reliability measured.

The Inter-Method Reliability section addressed the question of whether differences in familiarity with the methodology might influence inter-method reliability. In this section it was proposed that differences in familiarity with the methodology may indeed impact reliability.

8. OTHER RESULTS

Heuristics

The data collected during this study allowed calculating some useful heuristics. These heuristics are irrelevant to the study of reliability. They are included here because practitioners may find them useful. The following table summarizes two of the heuristics:

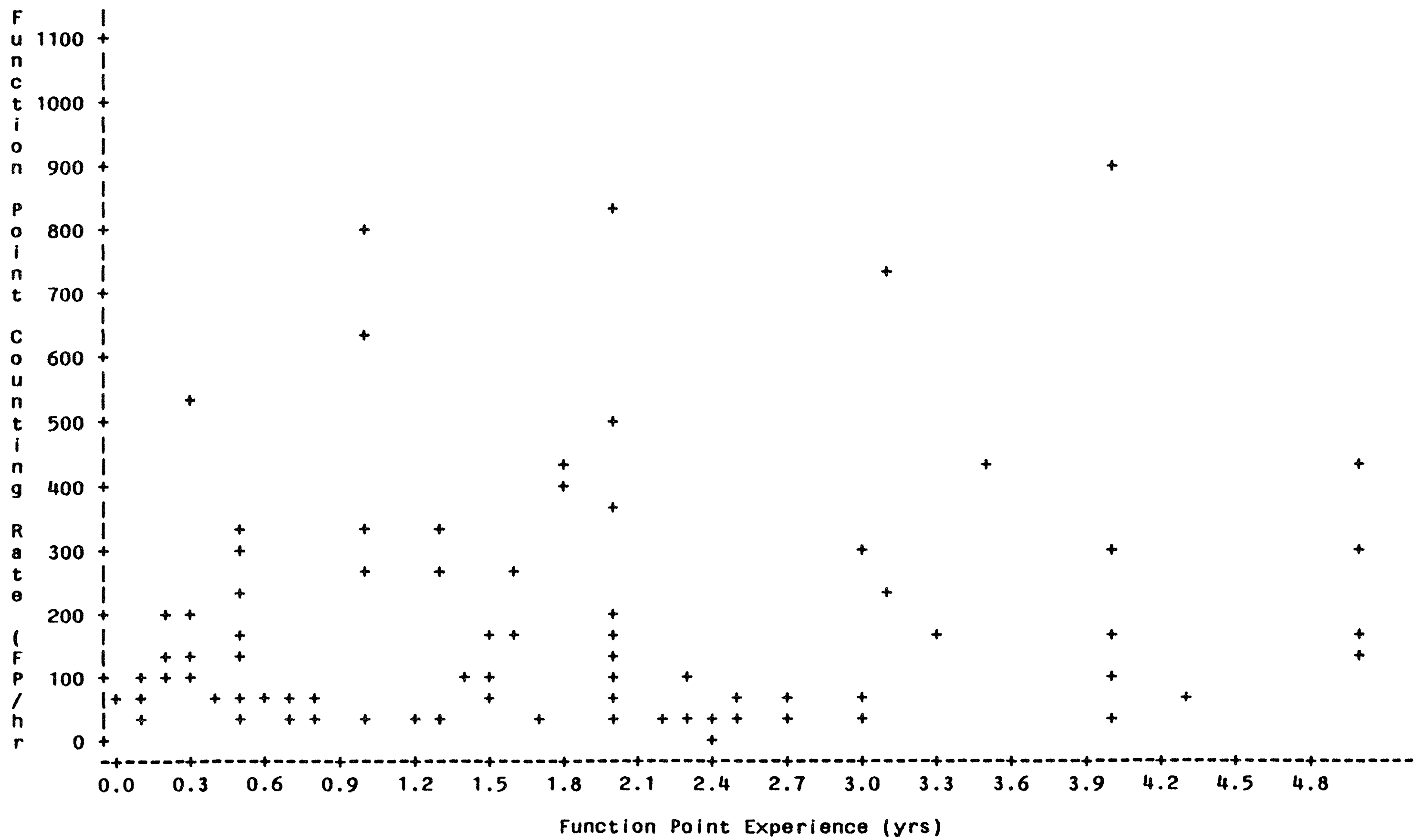
	Number of Data Points	Mean	Standard Deviation
Counting Rate (FP/Hr)	112	162	189
Work Effort per Function Point (Work-Hrs/FP)	13	16	11

A third heuristic also involves counting rates. It was hypothesized that raters with more Function Points experience would have higher counting rates. This was tested simply by plotting counting rate against experience. Figure 8.1 is this plot. The plot does not display any discernable trends.

Figure 8.1

Experience v. Counting Rate;

PLOT OF ANALRATE*FPCT_TIM   SYMBOL USED IS +



Profiles of Raters, Applications, and Sites

The questionnaires used in this study requested descriptive information about the raters, the applications, and the participating sites. This section tabulates the descriptive data collected. Discrete data (e.g. how many of the participating sites are part of a financial institution) are the tabulated on the questionnaires. Non-discrete data, (e.g. the distribution of experience levels) are summarized in the tables below.

The relevant appendices are:

J. Tabulation of Rater Backgrounds
K. Tabulation of Application Descriptions
M. Tabulation of Site Questions

The following table summarizes the non-discrete data:

Rater Backgrounds:

Item	Question Number	Number of Data Points	Mean	Standard Deviation
Application Development Experience (yrs)	1	81	10.9	7.4
Time with Current Employer (yrs)	2	80	6.1	6.4
Function Points Analyses Experience (yrs)	7	81	1.5	1.3
Function Points Training (days)	8a	58[6]	1.5	0.9

Site Questions:

Item	Question Number	Number of Data Points	Mean	Standard Deviation
Length of Time Site Has Used FPs (yrs)	2	42	2.6	1.8
Training Required Prior to Counting (d)	5	25	8.5	5.4
Pages of Internally Developed FP Rules	5	23	35.7	22.0
Contingency Added to Counts (%)	7a	9	26.7	11.7

Questions About the Count:

Item	Question Number	Number of Data Points	Mean	Standard Deviation
Time Spent (hrs)		114	4.6	3.4

[6]Includes only raters who have had formal training

Survey of Current Counting Practices

As was described in the Research Methodology, a portion of the data collection instrument included micro cases which asked participants to describe how they would count a particular situation. A total of 43 sites returned these micro-cases. The micro-cases were designed to determine what counting practices practitioners are using in certain aspects of Function Points analysis which were believed to be contentious.

Appendix L - Micro-Cases provides a tabulation of the questionnaire responses.

Participant Comments

Appendix N provides copies of general comments provided by participants on the questionnaires.

9. CONCLUSIONS

The key result of this study is that inter-rater reliability of the Function Point metric is quite high. On average, a Function Point count can be expected to within about 11% of an average count for the application.

A second result is that Function Points Analysis is robust across variants. At least as tested, counts produced with IFPUG 3.0 are within 10% of counts produced with an Entity-Relationship methodology.

The third result is less certain. The IFPUG 3.0 methodology appears to be more reliable than E-R methodology. However, this may be due to the IFPUG 3.0 methodology being more familiar to the participants in this study than was the E-R methodology.

REFERENCES

Albrecht, Allan J.; "Measuring Application Development Productivity"; in Proceedings of the IBM Applications Development Symposium, GUIDE/SHARE (Monterey, California, October 14-17);IBM; 1979; PP. 83-92.

Albrecht, Allan J. and Gaffney, John E.;, "Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation"; IEEE Transactions on Software Engineering, Vol. SE-9, No. 6; November 1983; pp. 639-648.

Albrecht, Allan J.; Personal correspondence; 1990.

Boehm, B.W.; "Improving Software Productivity"; Computer; September 1987; pp. 43-57.

Behrens, Charles A.; "Measuring the Productivity of Computer Systems Development Activities with Function Points"; IEEE Transactions on Software Engineering, Vol. SE-9, No. 6, November 1983; pp.648-652.

Conte, S., Dunsmore, H., and Shen, V.; Software Engineering Metrics and Models; Benjamin/Cummings, Menlo Park, CA; 1986.

Desharnais, Jean-Marc; Analyse Statistique de la Productivité des Projects de Développement en Informatique à Partir de la Technique des Points de Fonction; Universite du Quebec a Montreal; 1988.

Dreger, J. Brian; Function Point Analysis; Prentice Hall, Englewood Cliffs, NJ; 1989

Jones, T. Capers; Programming Productivity, McGraw-Hill, New York, 1986.

Kemerer, Chris F.; "An Empirical Validation of Software Cost Estimation Models"; Communications of the ACM; May 1987, Volume 30, Number 5; pp. 416-429.

Kemerer, Chris F.; Software Cost Estimation Models, in Software Engineers Reference Book, Butterworth Scientific Ltd, 1989.

Kirk, J. and Miller, M.; Reliability and Validity in Qualitative Research; Sage Publications, Newbury Park, CA; 1989.

Low, Graham C. and Jeffery, D. Ross; Function Points in the Estimation and Evaluation of the Software Process, IEEE Transactions on Software Engineering, Vol. 16 No. 1, January 1990; pp. 64-71.

Porter, Ben; unpublished draft of Entity-Relationship Modeling Methodology to be included in a future release of the IFPUG Function Point Counting Practices Manual; 1989.

Pressman, Roger S.; Software Engineering: A Practitioner's Approach, Second Edition; McGraw-Hill, 1987.

Ratcliff, Bryan and Rollo, Anthony L.; "Adapting Function Point Analysis to Jackson System Development"; Software Engineering Journal, pp. 79-84; January 1990.

Rudolph, Eberhard E.; "Productivity in Computer Application Development", Department of Management Studies, University of Auckland, 1983.

Rudolph, Eberhard E.; Personal correspondence; 1990.

Sprouls, Jack (ed.); IFPUG Function Point Counting Practices Manual Release 3.0; International Function Point Users Group, Westerville, OH; 1990.

Symons, Charles R.; Function Point Analysis: Difficulties and Improvements, IEEE Transactions of Software Engineering, Vol. 14, No. 1, January 1988; pp. 2-11.

Verner, J.M., Tate, G., Jackson, B. and Hayward, R.G.; "Technology Dependence in Function Point Analysis: A Case Study and Critical Review"; 11th International Conference on Software Engineering; 1989; pp. 375-382.

Zwanzig, Ken, editor; Handbook for Estimating Using Function Points; GUIDE Project DP-1234; published by GUIDE International; 1984.

APPENDICES

Appendix A - Letter to Outside Reviewers

Address

Dear *Outside Reviewer*:

IFPUG and the MIT Sloan School of Management are embarking on a study which will address many key issues in Function Point analysis. The study will answer the questions:

- How do organizations achieve high consistency in Function Point analyses?
- What counting conventions do IFPUG members prefer?
- How do Function Point analyses prepared with the proposed IFPUG Counting Manual 3.0 compare with analyses made with current counting practices?

In order to ensure this study's success, we are asking you and several other prominent people in the Function Point community to review the research design for this study. Any suggestions will be greatly appreciated, but we would specifically like you to address the questions:

- Are the instructions and questions clear?
- What do you feel are the principle causes of inter-rater and inter-company variability in Function Point analysis?
- In your experience, what is the average size of project in terms of both Function Points and work-months? What would be the standard deviation of such measures?
- Are there any additional questions that we should address?
- Would you have high confidence in the results of this study?

Due to IFPUG's tight deadline, please mail you suggestions to Professor Chris Kemerer at the above address within the next ten days. Alternatively, you can send us a facsimile at 617-258-7579.

Your input is greatly appreciated.

Sincerely,

Michael J. Connolley
Research Assistant to
Prof. Kemerer

Appendix B - Letter Soliciting Participation

Address

Dear *Site Contact*:

IFPUG and the MIT Sloan School of Management are embarking on a study which will address many key issues in Function Point analysis. To complete this valuable study we need your participation. The study will answer the questions:

+ How do organizations achieve high consistency in Function Point analyses?
+ What counting conventions do IFPUG members prefer?
+ How do counts prepared with the proposed IFPUG Counting Manual 3.0 compare with counts made with current counting practices?

Your responses will be kept strictly confidential.

The study results will help guide the IFPUG Counting Practices Committee in their efforts towards standardization. We expect to present preliminary results of the study at the IFPUG Spring Conference in April. Additionally, if you participate, the study will provide your site feedback which will enable you to determine:

+ How variable your site's Function Point analyses are relative to other IFPUG members' analyses?
+ What steps that you could take would have the greatest impact on minimizing variability?
+ How do your site's counting conventions compare with other IFPUG members' conventions.

What we need from your site are four independent Function Point analyses of two recently developed applications, and completion of a brief questionnaire. We expect that participating in the study will require two to four work-days. We will mail out full instructions and materials at the beginning of January, and we ask that you complete them by January 31.

We greatly encourage your participation. Please indicate you willingness to do so by completing the attached postcard with a "Yes" decision, and return it to IFPUG by

December 15th. If you have any questions, please call Andrew J. Belden at 203 382-4193.

Sincerely,

Bob Pickford
President

Appendix C - Cover Letter for Data Collection Instruments Package

Address

Dear *Site Contact*:

Thank you for participating in IFPUG's Function Point evaluation study which will address the following questions:

- How do organizations achieve high consistency in Function Point analyses?
- What counting conventions do IFPUG members prefer?
- How do counts prepared with the updated Albrecht '84 methodology compare with counts made with a data modeling methodology?

Your responses will be kept strictly confidential.

The study results will help guide the IFPUG Counting Practices Committee in their efforts towards standardization. Additionally, the study will provide your site feedback which will enable you to determine:

- How variable your site's Function Point analyses are relative to other IFPUG members' analyses?
- What steps you could take that would significantly impact your site's variability?
- How your site's counting conventions compare with other IFPUG members' conventions.

Attached are the following:

- Instructions for you, the site Coordinator
- Two copies each of two draft counting manuals: one based on the Albrecht '84 methodology, and the other based on data modeling methodology
- Four forms entitled "Form for Rater"
- A form entitled "Questionnaire for Coordinator"
- Five return envelopes

It is very important that all of the analyses are done independently. Please have each rater mail his/her completed form directly to the MIT research team as soon as it is complete. We need for you to have the questionnaire and all of the forms completed and mailed by January 31. If you have any questions, please call the IFPUG project manager, Andrew J. Belden, 203-382-4193.

Sincerely,

Robert C. Pickford

Appendix D - Instructions for the Coordinator

Introduction.
There are three basic steps to this study. You will need to select two applications; have four people make Function Point analyses of both applications; and, lastly, complete a questionnaire. More detailed instructions are provided below.

This packet contains the following:
- This page of instructions
- Two copies each of two draft counting manuals: one based on the Albrecht '84 methodology, and the other based on data modeling methodology
- Four forms entitled "Form for Rater"
- A form entitled "Questionnaire for Coordinator"
- Five return envelopes

Part 1 - Analyses of In-House Systems.
1. Please select two medium-sized software applications for which your site will be able to prepare complete Function Point analyses. These will be referred to as "Application 1" and "Application 2". By "medium-sized", we mean an application that required between one and six work-years to develop.
2. Please select four people (one of which may be you) to prepare Function Point analyses. These people will be referred to as "Rater A", "Rater B", "Rater C", and "Rater D".
3. Have Rater A and Rater B prepare Function Point analyses of both Application 1 and Application 2. These analyses must be made using the updated Albrecht '84 counting manual which we provided. Raters A and B should use the same basis when making their analyses. Preferably, this basis should be a requirements definition type of document (e.g. entity-relationship diagrams and process flow diagrams). The raters must work independently.
4. Have Rater C and Rater D prepare Function Point analyses of both Application 1 and Application 2. These analyses must be made using the data modeling counting manual which we provided. Raters C and D should also use the same basis when making their analyses. Preferably, this basis should be a requirements definition type of document (e.g. entity-relationship diagrams and process flow diagrams). The raters must work independently.
5. Have the all the Raters complete their respective forms for each application (eg. Rater A completes the form titled "Form for Rater A"). As soon a Rater has completed his form, he should place it in one of the pre-addressed envelopes and mail it to the MIT research team at the following address:

Function Points Evaluation Study
c/o Prof. Chris F. Kemerer
MIT E53-315
Sloan School of Management
50 Memorial Drive
Cambridge, MA 02139

6. When all the analyses are completed, please destroy the draft counting manuals we provided.

Part 2 - Questionnaire and Micro Cases. For part 2, please answer the questions on the form titled "Questionnaire for Coordinator." This questionnaire should take approximately one hour to complete.

Appendix E - Methodology I, the IFPUG 3.0 Methodology

FUNCTION POINTS COUNTING METHODOLOGY I

For Use By Raters A and B

I. FUNCTION POINT ANALYSIS OVERVIEW

A. Objectives of Function Point Analysis

Function Points measure software by quantifying the functionality provided external to itself, based primarily on logical design. With this in mind, the objectives of Function Point counting include:

- o provide a normalization factor for software comparison;
- o provide a sizing metric allowing for productivity analysis;
- o measure independently of technology used for implementation;
- o provide a vehicle for software estimation;
- o measure what the user requested and received;

In addition to meeting the above objectives, the process of counting function points should be:

- o simple enough to minimize the overhead of the measurement process;
- o simple yet concise, to allow for consistency over time, projects and practitioners.

B. Summary of Function Point Counting

The Function Point metric measures an application based on two areas of evaluation. The first results in the Unadjusted Function Point count and reflects the functionality provided to the user by the application. The second area of evaluation, which produces the Value Adjustment Factor (VAF), evaluates the overall complexity of the application.

The Function Point metric measures the application in terms of WHAT is delivered not HOW it is delivered. Only user-requested and visible components are counted. These components called Function Types and are categorized as either Data or Transactional.

Function Types:

Data:

Internal Logical Files (ILF) - internally maintained logical group of data.
External Interface Files (EIF) - externally maintained logical group of data.

Transactional:

External Inputs (EI) - maintains internally stored data.
External Outputs (EO) - standardized data output.
External Inquiries (EQ) - on-line combination of input (request) and output (retrieval).

Each Function Type is further categorized based on its complexity.

- Low
- Average
- High

Complexity rating criteria vary between Function Type, and is discussed in Section IV, Function Point Calculation.

Function Point values, ranging from 3 to 5 depending on the Function Type and complexity rating, are assigned and totaled producing the Unadjusted Function Point Count.

The resulting number is then weighted by the Value Adjustment Factor (VAF) to produce the Adjusted Function Point Count. The VAF is comprised of 14 General System Characteristic (GSC) questions. The GSC questions assess overall system requirements.
The General System Characteristics are:

1. Data Communication
2. Distributed function
3. Performance
4. Heavily use configuration
5. Transaction rates
6. On-line data entry
7. Design for end user efficiency
8. On-line update
9. Complex processing
10. Usable in other applications
11. Installation ease
12. Operational ease
13. Multiple sites
14. Facilitate change

The questions are answered using Degrees of Influence (DI) on a scale of 0 to 5.

0 Not Present, or no influence
1 Incidental influence
2 Moderate influence
3 Average influence
4 Significant influence
5 Strong influence throughout

C. Types of Function Point Counts

Function Point counts can be associated to either projects or applications. There are three types of function point counts:

1. An Application Function Point Count reflects the installed functionality for an application. Also called a Baseline or Installed Function Point Count, it reflects the installed functionality of the application.

2. Development (Project) Function Point Count - Function Point count associated with the initial installation of new software. This count reflects the installing project's user functionality.

3. Enhancement (Project) Function Point Count - Function Point count associated with the enhancement of existing software. An Enhancement Function Point Count reflects those modifications to an existing application which add, change or delete user functionality.

D. Boundaries

Boundaries identify the border between the application / project being measured and either, external applications or the user domain.

Boundaries are used to establish the scope of the work product being measured. Additionally, they are used to establish data ownership and processing relationships, required when conducting a function point count.

Application Boundary. Look at the application from the POINT OF VIEW OF THE USER, what the user can touch and feel. Use the system external specs or get a system flow chart and draw a boundary around it to highlight what is internal vs. what is external to the application.

One time requirements, e.g. conversions, which are included as part of Development and Enhancement projects are considered external to an application's boundary.

Development (Project) Boundary. Again, look at the application from the POINT OF VIEW OF THE USER, what the user can touch and feel. Use the system external specs or get a system flow chart and draw a boundary around it to highlight what is internal vs. what is external to the application.

A Development project boundary includes any data conversion requirements.

Enhancement (Project) Boundary. An Enhancement project must conform to the boundaries already established for the application(s) being modified. An Enhancement project boundary includes any data conversion requirements.

II. COUNTING RULES

General

Function Points are calculated based on user-requested, visible components of an application. These components are called Function Types and are categorized as either Data or Transactional.
Data Function Types evaluate the functionality provided the user for internal and external data requirements.

Internal Logical Files (ILF) reside internal to an application's boundary and reflect data storage functionality provided to the user. Internal Logical Files must be maintained and utilized by the application.

External Interface Files (EIF) reside external to an application's boundary and reflect the functionality provided by the application through the use of data maintained by other applications.
While both Internal Logical Files and External Interface Files contain the word "file" in their title, they are not files in the traditional DP sense of the word. In this case, file refers to a logically related group of data and not the physical implementation.

Transactional Function Types evaluate the functionality provided the user for the processing requirements of an application.

External Inputs (EI) reflect the functionality provided the user for the receipt of and maintenance of data on Internal Logical Files.

External Outputs (EO) reflect the functionality provided the user for output generated by the application.

External Inquiries (EQ) reflect the functionality provided the user for on-line queries of Internal Logical Files or External Interface Files.

A. Internal Logical Files

Overview. An application's maintainable data storage requirements are evaluated and contribute to the function point count based on the number and complexity of Internal Logical Files.

Definition. An Internal Logical File (ILF) is a user identifiable group of logically related data or control information maintained and utilized within the boundary of the application.

User identifiable group of logically related data is defined as: logically related data, related at such a level that an experienced user would identify the data as fulfilling a specific user requirement of the application. The Data Analysis equivalent to such high level logical groupings are singularly named data stores on a data flow diagram.

Control information is data used by the application for control purposes, ie., meeting Business Function requirements.

Maintained is the ability to add, modify or delete data through a standardized process of the application.

Identification.

1. Identify all data which is:
   a) stored internal to the application's boundary.
   b) maintained through a standardized process of the application.
   c) identified as a requirement of the application by the users.

2. Group the data logically based on the user's view.
   a) This grouping should occur at the level of detail at which the user can first categorize the data as satisfying unique requirements of the application.
   b) View the data logically. While some storage technologies will relate closely to Internal Logical Files, eg. tables in a relational DBMS or a sequential flat file, a one to one relationship should NOT be assumed.

Examples. When identifying potential ILFs, do not look at the storage technology, eg. tables, flat file, indexes, paths, etc. Instead, look at the type of data stored, and how a user would view/group the data. Each type of data on the following list can relate to one or more ILFs, depending on the user's view.

- application data - (master files), eq. Tax information, Personnel information, etc.
- application security data
- audit data
- help messages
- error messages
- edit data
- backup data - Backup data should be counted ONLY if specifically requested by the user due to legal or similar requirements.

- Internal Logical Files maintainable by more than one application should be credited to both applications at the time each is counted.

The following are not ILFs:

- temporary files
- work files
- sort files
- suspense files - Files containing incomplete transactions from an External Input. Do not count unless data on the suspense file can be accessed and maintained by the user through a unique External Input.
- backup data - required due to corporate Backup and Recovery procedures.
- those files introduced only because of technology used, eg. a file containing JCL required for job submission.

Additional Guidance.

- Internal Logical Files having more than 100 fields should be reviewed and where appropriate, subdivided into multiple ILFS. A procedure which can aid in dividing a large Internal Logical File into multiple ILFS is called Super Files. See Appendix A for a discussion of Super Files.
- backup file - Backup files should be counted ONLY if specifically requested by the user due to legal or similar requirements. Backup files required due to normal Backup and Recovery procedures are not counted.
- LIFs maintainable by more than one application. LIFs maintainable by more than one application should be credited to both applications at the time each is counted.
- Suspense files - Suspense/Carry Around files should be counted as a Internal Logical File ONLY if the suspense file can be updated by the user through a separate External Input.

B. External Interface Files

Overview. External Interface Files represent an application's externally maintained data storage requirements. External Interface Files are evaluated and contribute to the function point count based on their number and complexity.

Definition. An External Interface File (EIF) is a user identifiable group of logically related data or control information utilized by the application which is maintained by another application.

User identifiable group of logically related data is defined as: logically related data, related at such a level that an experienced user would identify the data as fulfilling a specific user requirement of the application. The Data Analysis equivalent to such high level logical groupings are singularly named data stores on a data flow diagram.

Control information is data used by the application to insure compliance with Business Function Requirements specified by the user.

Maintained External is defined as the maintenance of data performed by another application.

Identification.

1. Identify all data which is:
   a) stored external to the application's boundary.
   b) NOT maintained by this application.
   c) identified as a requirement of the application by the users.

2. Group the data logically based on the user's view.
   a) This grouping should occur at the level of detail at which the user can first categorize the data as satisfying unique requirements of the application.
   b) View the data logically. While some storage technologies will relate closely to External Interface Files, eq. tables in a relational DBMS or a sequential flat file, a one to one relationship should NOT be assumed.

Examples. When identifying potential EIFs, do not look at the storage technology, eg. tables, flat files, indexes, paths, etc. Instead, look at the type of data and how a user would view it. Each type of data on the following list can relate to one or more EIFs, depending on the user's view.

- reference data - external data which is utilized by the application, but NOT maintained on Internal Logical Files.
- application security data
- help messages
- error messages
- edit data

The following are not EIF:

- data received from another application which adds, changes or deletes a to ILF. This would be considered transaction data and therefore be counted as an External Input.
- data maintained by the application being counted which is accessed and utilized by another application. Data which is formatted and processed for use by another application should be counted as an External Output.

Additional Guidance:

- EIF is not credited to the "sending application" regardless of whether a file is counted as an External Interfaces / External Outputs or External Interfaces / External Inputs, function type determination is based on how the application which did not originate the data utilizes the data. If the data is viewed as transactional, ie., used to update a Internal Logical File, it is either an External Input or External Output, depending on data flow. If viewed as reference data, ie., does not update a Internal Logical File, it is an External Interface regardless of the data flow.

- Problems arise when the application being function point counted tries to differentiate between External Interfaces and External Outputs. Identification is straightforward when it is known how the other application is using the data, ie., transactional or reference. Inconsistencies arise when it is not known how the other application utilizes the data. Various interpretations have evolved. Regardless of the interpretation used, which in themselves create inconsistencies, varying counts will result depending on whether or not the application being counted knows how the other application utilizes the data.

C. External Inputs

Overview. External Inputs represent an application's data maintenance and control processing requirements. External Inputs are evaluated and contribute to the function point count based on their number and complexity.

Definition. An External Input (EI) processes data or control information which enters the application's external boundary and through an unique logical process maintains a Internal Logical File. An External Input should be considered unique if it has a different format, or if the logical design requires processing logic different from other External Inputs of the same format.

An External Input is considered unique if:

1. the input format is unique; and data stored on an Internal Logical File is maintained. or
2. the processing logic is unique; and data stored on an Internal Logical File is maintained.

Control information is data used by the application to insure compliance with Business Function Requirements specified by the user.

Maintain is the ability to add, change or delete data through a standardized process of the application.

Format is defined as unique data elements or an unique arrangement/order of data elements.

Processing Logic is defined as unique edits, calculations / algorithms and/or sorts specifically requested by the user.

Identification.

1. Identify all processes which update a Internal Logical File.
2. For each process identified: a) if the data used by the process can be received in more than one format, each format is considered a separate process. b) credit an External Input for each data maintenance activity performed, ie., add, change and delete.

Examples. The following are External Inputs, assuming the above conditions are met:

- transactional data - external data which is used to maintain Internal Logical Files.
- screen input - Count one External Input for each process which maintains a ILF. If Add, Change and Delete capabilities are present, the screen would count as 3 External Inputs. (Screen in this context is a logical screen which can be one or more physical screen(s) processed as one transaction. Conversely, one physical screen, when viewed by processes, can relate to multiple External Inputs.)

- batch input. For each unique process which maintains a ILF, count one External Input for each process, ie., Add, Change and Delete. Batch inputs should be identified based on the processing required to apply the data. One physical input file can, when viewed logically, correspond to a number of External Inputs. Conversely, 2 or more physical input files can correspond to one External Input, if the processing logic and format are identical for each physical file. A method for identifying multiple External Inputs from the processing of one physical file is to look at the record types on the file. Exclude header and trailer records as well as those record types required due to physical space limitations. (Do not exclude header and trailer records from this process if due to user requirements, they are required for audit purposes.) Review the remaining record types for unique processing requirements and associate an External Input for each unique process. Do not assume a one to one correspondence between the remaining record types and External Inputs.
- Duplicate External Inputs - Input processes which if specifically requested by the user, duplicate a previously counted External Input, should each be counted. An Example being: A banking system which accepts identical deposit transactions, one through an Automated Teller Machine (ATM) transaction and a second through a manual teller deposit transaction.
- Input processes which maintain either a Internal Logical File or a suspense/-carry around file depending on edit evaluation, should be counted based on the following: If the suspense/carry around file is maintainable by the user, the suspense/carry around file is counted as a Internal Logical File. That being the case, count External Inputs for each data maintenance activity performed on both Internal Logical Files . If the suspense/carry around file is not maintainable by the user, count External Inputs for each data maintenance activity performed on the original Internal Logical File. In either instance, the process of reapplying data from the suspense/carry around file to the Internal Logical File is NOT counted.

The following are not External Inputs:

- reference data - external data utilized by the application, but NOT maintained on Internal Logical Files.
- Input side of an External Inquiry, data input used to drive selection for data retrieval.
- Menu screens - screens which provide only selection or navigational functionality and do NOT maintain a Internal Logical File are not counted.
- Logon Screen - screens which facilitate entry into an application and do NOT maintain a Internal Logical File are not External Inputs.
- Multiple methods of invoking the same input logic, eg., entering "A" or "Add" on a Command line, or a PF key, should be counted only once.
- Logon Screen: Screens which facilitate entry into an application and do NOT maintain an Internal Logical File are not External Inputs.

D External Output

Overview. External Outputs represent an application's output processing requirements. External Outputs are evaluated and contribute to the function point count based on their number and complexity.

Definition. An External Output (EO) processes data or control information which EXITS the application's external boundary. An External Output should be considered unique if it has a different format, or if the logical design requires processing logic different from other External Outputs of the same format.

An External Output is considered unique if:
1) the output format is unique, or
2) the processing logic is unique.

Control information is data used by the application to insure compliance with Business Function Requirements specified by the user.

Format is defined as unique data elements or an unique arrangement/order of data elements.

Processing Logic is defined as unique edits, calculations / algorithms and/or sorts specifically requested by the user.

Identification.

1. Identify all processes which:
   a) send data external to the application's boundary. or
   b) send control data external to the application's boundary.

2. For each process identified:
   a) if the data used by the process is sent in more than one format, each format is considered a separate process.
   b) credit an External Output for each process.

Examples. The following are External Outputs, assuming the above conditions are met:

- data transfer to other applications. Data residing on a LIF which is formatted and processed for use by an external application. Outputs should be identified based on the processing required to manipulate the data.One physical output file can, when viewed logically, correspond to a number of External Outputs. Conversely, 2 or more physical output files can correspond to one External Output, if the processing logic and format are identical for each physical file. Instances where multiple External Outputs result from one physical output file can be identified through the existence, on the output file, and use during processing, of record types, eg., op codes, command codes and transaction codes, etc.
- reports - Each report produced by the application should be counted as an External Output. Two identically formatted reports, at the detail and

summary levels would be counted as 2 External Outputs. Each report requires unique processing logic, i.e., unique calculations.

- duplicate reports - Identical reports, produced on different media, due to specific user requirements, are separate External Outputs, since they are considered to have different processing logic, eg., identical reports one on paper, one on microfiche.
- on-line reports - on-line output of data which is NOT an immediate response to an input.
- error/confirmation messages - An External Output should be credited for each EXTERNAL INPUT having error/confirmation messages. Complexity is based on the total number of specific error/confirmation messages, ie. DETs, which are available to the screen(s). Implementation techniques, whether a Message Area/Box/Window appearing on the External Input, or a separate physical screen, eg., Message Frame, do not impact the complexity or number of External Outputs associated to a particular External Input.

The following are not External Outputs:

- help - see External Inquiry.
- Multiple methods of invoking the same output logic, eg., entering "A" or "Add" on a Command line, or a PF key, should be counted only once.
- Error/Confirmation messages associated to function types other than External Inputs. For example, an External Output would NOT be counted for error/confirmation messages associated to an External Inquiry.
- Multiple reports/unique data values - Identical reports which have the same format and processing logic, but exist due to unique data values are NOT counted as separate External Outputs. For example, two reports which are identical in format and processing logic, the first containing customer names beginning with "A" through "L" and the second having customer names beginning with "M" through "Z", would be counted as only one External Output.
- Summary information on a detail report does not constitute a unique External Output.
- Area/Box/Window appearing on the External Input or a separate physical screen, e.g., Message Frame, do not impact the complexity or number of External Outputs associated to a particular External Input.

E. External Inquires

Overview. External Inquiries represent an application's inquiry processing requirements. External Inquiries are evaluated and contribute to the function point count based on their number and complexity.

Definition. An External Inquiry (EQ) is a unique input/output combination where an input causes an immediate output and a LIF is not updated. An External Inquiry should be considered unique if it has a format different from other External Inquiries in either its input or output parts, or if the external design requires processing logic different from other external inquiries.

An Input/Output combination is considered unique if:

1) the input format is unique or
2) the processing logic is unique or
3) the output format is unique.

Format is defined as unique data elements or an unique arrangement/order of data elements.

Processing Logic is defined as unique edits, calculations / algorithms and/or sorts specifically requested by the user.

Identification.

1) Identify all processes which: a) an input triggers an immediate retrieval of data.

2) For each process identified: a) verify that each input/output combination is unique, each unique input/output combination is considered a separate process. b) credit an External Inquiry for each process.

Examples.

The following are External Inquiries, assuming the above conditions are met:

- an immediate on-line retrieval of data, selection of which is based on data input.
- implied inquiries - change/delete screen(s) which provide data retrieval capabilities prior to change/delete functionality should be credited with an External Inquiry. If the input and output sides of the External Inquiry are identical for both change and delete functions, count only one External Inquiry.
- Menus having implied inquiries - Menu screens which provide screen selection and data selection criteria for the called screen should be counted as External Inquiries.
- Logon screens - Logon screens which provide security functionality should be counted as an External Inquiry.
- help - is defined as an inquiry pair where the input and the output (explanatory text) are both unique. Credit only once, help text which can be

accessed or displayed through different request techniques, (PF keys, Command line entry, cursor positioning, different calling screen), or from different areas of an application, ie., the identical Help text is retrieved from an application screen or a Help subsystem.

There are three categories of Help which are considered External Inquiries:

1) Full Screen Help - A help facility which dependent on the application screen, displays help text relating to the calling screen. Credit one low complexity External Inquiry per calling screen, regardless on the number of Help panels/screens returned.
2) Field Sensitive Help - A help facility which dependent on the location of the cursor or some other identifying method, displays help documentation specific to the field. Credit an External Inquiry per screen, complexity based on the number of fields (DETs) which are field sensitive.
3) Help Subsystems - Help facility which can be accessed and browsed independent of the associated application. If the text retrieved from a Help Subsystem is identical to Full Screen Help, do not count the Help Subsystem. Specific counting rules for Help Subsystems are not available at this time.

The following are not External Inquires:

- Error/Confirmation Messages - See External Outputs.
- Multiple methods of invoking the same inquiry logic, eg., entering "A" or "Add" on a Command line, or a PF key, should be counted only once.
- Help text which can be accessed from multiple areas/screens of an application should be counted only once.
- Menu Screen(s) - Menu screens which provide only screen selection functionality are not counted.

Additional Guidance:

- Menus having implied inquiries: Menu screens which provide screen selection AND data selection criteria for the called screen should be counted as External Inquiries.
- Menu Screen(s) - Menu screens which provide only screen selection functionality are not counted.
- Logon screens: Logon screens which provide security functionality should be counted as an External Inquiry.
- Help is defined as an inquiry pair where the input and the output (explanatory text) are both unique. Credit only once, help text which can be accessed or displayed through different request techniques, (PF keys, Command line entry, cursor positioning or different calling screen), or from different areas of an application, i.e., the identical Help text is retrieved from an application screen or a Help subsystem.

III. GENERAL SYSTEMS CHARACTERISTICS

Each General System Characteristic (GSC) must be evaluated in terms of its Degree of Influence (DI) on a scale of 0 to 5. The descriptions listed under "Score as:" are meant to be guides in determining the Degree of Influence. If none of the guideline descriptions fit the application exactly, a judgement must be made about which Degree of Influence most closely applies to the application.

1. Data Communications

The DATA and CONTROL information used in the application are sent or received over communication facilities. Terminals connected locally to the control unit are considered to use communication facilities. Score as:

0 - Application is pure batch processing or a stand alone PC.

Remote data entry/printing

1 - Application is batch but has remote data entry or remote printing.

2 - Application is batch but has remote data entry and remote printing.

Interactive teleprocessing (TP)

3 - On-line data collection or TP front end to a batch process or query system.

4 - More than a front-end, but the application supports only one type of TP communications protocol.

5 - More than a front-end, but the application supports more than one type of TP communications protocol.

2. Distributed Data Processing

Distributed data or processing functions are a characteristic of the application within the application boundary. Score as:

0 - Application does not aid the transfer of data or processing function between components of the system.

1 - Application prepares data for end user processing on another component of the system (e.g. PC spreadsheet, PC DBMS, etc.).

2 - Data is prepared for transfer, is transferred, and is processed on another component of the system (not for end user processing).

3 - Distributed processing and data transfer is on-line and in one direction, only.

4 - Distributed processing and data transfer is on-line and in both directions.

5 - Processing functions are dynamically performed on the most appropriate component of the system.

3. Performance

Application performance objectives, stated or approved by the user, in either response or throughput, influenced (or will influence) the design, development, installation and support of the application. Score as:

0 - No Special performance requirements were stated by the user.

1 - Performance and design requirements were stated and reviewed but no special actions were required.

2 - On-line response time is critical during peak hours. No special design for CPU utilization was required. Processing deadline is for the next business day.

3 - On-line response time is critical during all business hours. No special design for CPU utilization was required. Processing deadline interface systems is not constraining.

4 - Stated user performance requirements are stringent enough to require performance analysis tasks in the design phase.

5 - In addition, performance analysis tools were used in the design, development, and/or implementation phases to meet the stated user performance requirements.

4. Heavily Used Configuration

A heavily used operational configuration, requiring special design considerations, is a characteristic of the application (e.g., the user wants to run the application on existing or committed equipment that will be heavily used). Score as:

0 - No explicit or implicit operational restrictions.

1 - Operational restrictions do exist, but are less restrictive than a typical application. No special effort is needed to meet the restrictions.

2 - Some security or timing considerations.

3 - Specific processor requirements for a specific piece of the application.

4 - Stated operation restrictions require special constraints on the application in the central processor, or, a dedicated processor.

5 - In addition, there are special constraints on the application in the distributed components of the system.

5. Transaction Rate

The transaction rate is high and it influenced the design, development, installation, and support of the application. Score as:

0 - No peak transaction period anticipated.

1 - Monthly peak transaction period anticipated.

2 - Weekly peak transaction period anticipated.

3 - Daily peak transaction period anticipated.

4 - High transaction rates stated by the user in the application requirements or service level agreements are high enough to require performance analysis tasks in the design phase.

5 - High transaction rates stated by the user in the application requirements or service level agreements are high enough to, in addition, require the use of performance analysis tools in the design, development, and/or installation phases.

6. On-Line Data Entry

On-line data entry and control functions are provided in the application. Score as:

0 - All transactions are processed in batch mode.

1 - 1% - 7% of transactions are interactive data entry.

2 - 8% - 15% of transactions are interactive data entry.

3 - 16% - 23% of transactions are interactive data entry.
4 - 24% - 30% of transactions are interactive data entry.
5 - Over 30% of transactions are interactive data entry.

7. End User Efficiency

The on-line functions provided emphasize a design for end user efficiency. Efficiency can be delivered in one or more of the following forms:

- menus
- on-line help/documentation
- scrolling
- remote printing (via on-line transactions)
- preassigned function keys
- submission of batch jobs from on-line transactions
- cursor selection of screen data
- heavy use of reverse video, highlighting, colors underlining, etc.
- hard copy user documentation of on-line transactions
- mouse interface

- pop-up windows
- as few screens as possible to accomplish a business function
- easy navigation between screens (e.g., through function keys)
- bi-lingual support (supports 2 languages; count as 4 items)
- multi-lingual support (supports more than 2 languages; count as 6 items)

Score as:

0 - none of the above

1 - 1 - 3 of the above.

2 - 4 - 5 of the above.

3 - 6 or more of the above, but there are no specific user requirements related to efficiency.

4 - 6 or more of the above, and, stated requirements for end-user efficiency are strong enough to require design tasks for human factors to be included (e.g., minimize key strokes, maximize defaults, use of templates, etc.).

5 - 6 or more of the above, and, stated requirements for end-user efficiency are strong enough to require use of special tools and processes in order to demonstrate that the objectives have been achieved.

8. On-Line Update

The application provides on-line update for the internal logical files. Score as:

0 - None.

1 - On-line update of 1 to 3 control files. Volume of updating is low and recovery is easy.

2 - On-line update of 4 or more control files. Volume of updating is low and recovery is easy.

3 - On-line update of major internal logical files.

4 - In addition, protection against data lost is essential and has been specially designed and programmed in the system.

5 - In addition, high volumes bring cost considerations into the recovery process. Highly automated recovery procedures with minimum of operator intervention.

9. Complex Processing

Complex processing is a characteristic of the application. Categories are:

- Sensitive control (e.g., special audit processing) and/or application specific security processing.
- Extensive logical processing.
- Extensive mathematical processing.
- Much exception processing resulting in incomplete transactions that must be processed again; e.g., incomplete ATM transactions caused by a line drop.
- Complex processing to handle all input/output possibilities; e.g., multi-media, device independence, etc.

Score as:

0 - None of the above.

1 - Any one of the above.

2 - Any two of the above.

3 - Any three of the above.

4 - Any four of the above.

5 - Any five of the above.

10. Reusability

The application, and the code in the application, have been specifically designed, developed, and supported to be usable in other applications. Score as:

0 - No reusable code

1 - Reusable code is used within the application.

2 - Less than 10% of the modules produced considered more than one users needs.

3 - More than 10% of the modules produced considered more than one users needs.

4 - The application was specifically packaged and/or documented to ease re-use, and application is customized by user at source code level.

5 - The application was specifically packaged and/or documented to ease re-use, and applications is customized for use by means of parameter input.

11. Installation Ease

Conversion and installation ease are characteristics of the application. A conversion and installation plan and/or conversion tools were provided, and were tested during the system test phase. Score as:

0 - No special considerations stated by user and no special set up required for installation.
1 - No special considerations stated by user BUT special set up required for installation.

2 - Conversion and installation requirements were stated by the user and, conversion and installation guides were provided, and tested. The impact of conversion on the project is not considered to be important.

3 - Conversion and installation requirements were stated by the user and, conversion and installation guides were provided, and tested. The impact of conversion on the project is considered to be important.

4 - In addition to (2), automated conversion and installation tools were provided and tested.

5 - In addition to (3), automated conversion and installation tools were provided and tested.

12. Operational Ease

Operational ease is a characteristic of the application. Effective start-up, back-up, and recovery procedures were provided, and were tested during the system test phase. The application minimizes the need for manual activities, such as tape mounts, paper handling, and direct on-location manual intervention. Score as:

0 - No special operational considerations other than the normal backup procedures, were stated by the user.

1 to 4 - Select which of the following items apply to the application. Each item has a point value of one, except as noted otherwise.

- Effective startup, backup, and recovery processes were provided but operator intervention is required.
- Effective startup, backup, and recovery processes were provided but operator intervention is required. (count as 2 items).
- The application minimizes the need for tape mounts.
- The application minimizes the need for paper handling.

5 - Application is designed for unattended operation. Unattended operation means NO OPERATOR INTERVENTION required to operate the system other than to start up or shut down the application. Automatic error recovery is a feature of the application.

13. Multiple Sites

The application has been specifically designed, developed, and supported to be installed at multiple sites for multiple organizations. Score as:

0 - No user requirement to consider the needs of more than one user site.

1 - Needs of multiple sites were considered in the design and the application is designed to operate only under identical hardware and software environments.

2 - Needs of multiple sites were considered in the design and the application is designed to operate only under similar hardware and/or software environments.

3 - Needs of multiple sites were considered in the design and the application is designed to operate under different hardware and/or software environments.

4 - Documentation and support plan are provided and tested to support the application at multiple sites and application is as described by (1) or (2).

5 - Documentation and support plan are provided and tested to support the application at multiple sites and application is as described by (3).

14. Facilitate Change

The application has been specifically designed, developed, and supported to facilitate change. Examples are:

- Flexible query capability is provided.
- Business control data is grouped in tables maintainable by the user.

Score as:

0 - No special user requirement to design the application to minimize or facilitate change.

1 to 5 - Select which of the following items apply to the application. Each item has a point value of one except as noted otherwise.

- Flexible query facility is provided which can handle simple query requests; e.g., ANDs or ORs logic applied to only one internal logical file.
- Flexible query facility is provided which can handle query requests of average complexity: e.g., ANDs or ORs logic applied to more than one internal logical file . (count as 2 items)
- Flexible query facility is provided which can handle complex query requests: e.g., AND/OR logic combinations on one or more internal logical files. (count as 3 items)

- Control data is kept in tables that are maintained by the user with on-line interactive processes but changes take effect only on the next business day.
- Control data is kept in tables that are maintained by the user with on-line interactive processes and the changes take effect immediately. (count as 2 items).

IV. FUNCTION POINT CALCULATION

The Function Point calculation is a three step process. Step 1 produces the Unadjusted Function Point count, step 2 produces the Value Adjustment Factor (VAF) and step 3 adjusts the Unadjusted Function Point count by the VAF to produce the final Function Point count.

The formula used in step 3 varies depending on the type of count, Application (System Baseline), Development (Project) or Enhancement (Project). Sections 6.3, 6.4 and 6.5, discuss the final Function Point calculation for each type of count.

A. Unadjusted Function Point Calculation - Step 1

Unadjusted Function Point values are associated to each identified Function Type based on complexity. There are two types of Function Types, Data and Transactional.

Data Function Types include: Internal Logical Files and External Interface Files. The complexity of each identified Data Function Type is based on the number of Record Types (RET) and Data Element Types (DET).

Transactional Function Types include: External Inputs, External Outputs and External Inquiries. The complexity of each identified Transaction Function Type is based on the number of File Type Referenced (FTR) and Data Element Type (DET).

Once an application's components (specific data and processing requirements) have been categorized into the various function types, each component is assigned an unadjusted Function Point value based its complexity.

The Unadjusted Function Point value for each component is then summarized at the function type level and again at the application level. The resulting total at the application level is the application's Unadjusted Function Point count and is used later in the final calculation.

Internal Logical Files. Each identified Internal Logical File is assigned a complexity based on the number of Record Element Types (RET) and Data Element Types (RET) associated.

Record Element Type Identification

Record Element Types (RET) are sub groupings of Internal Logical Files based on the logical/user view of the data. The Data Analysis equivalent to such logical groupings are Data Entities on a data flow diagram. Internal Logical Files which cannot be sub categorized are considered to have one RET.

Data Element Type Identification

An Internal Logical File's Data Element Types (DET) are user recognizable, non-recursive fields residing on the Internal Logical File.

Each field on the Internal Logical File is a DET with the following exceptions:

- Fields should be viewed at the user recognizable level. For example, an account number or date which is physically stored in multiple fields should be counted as one DET.
- Fields which because of technology or implementation techniques appear more than once in an Internal Logical File should be counted only once. For example, if an Internal Logical File is comprised of more than one table in a relational DBMS, the keys used to relate the tables would be counted only once.
- Repeating fields which are identical in format and exist to allow for multiple occurrences of a data value, are counted only once. For example, an Internal Logical File containing 12 monthly budget amount fields and an annual budget amount field would be credited with 2 DETs, a DET for the monthly budget amount fields and a DET for the annual budget amount field.

Complexity assignment for Internal Logical Files (ILFs) is based on the following matrix:

	1 to 19 DET	20 to 50 DET	51 or more DET
1 RET	L	L	A
2 to 5 RET	L	A	H
6 or more RET	A	H	H

Legend
RET = Record Element Type
DET = Data Element Type (field)

Complexity
L = Low
A = Average
H = High

Use the following table to translate an Internal Logical File's complexity to Unadjusted Function Points.

Complexity Rating	Unadjusted Function Points
L (ow)	7
A (verage)	10
H (igh)	15

For example:

To calculate the Unadjusted Function Point count for a Internal Logical File having 23 DETs and 3 RET, you would use the ILF Complexity Matrix and find that the ILF is of Average Complexity. Secondly, using the ILF Unadjusted Function Point Table, transcribe the Average Complexity Rating to an Unadjusted Function Point value of 10.

An application's Internal Logical Files contribution to the total Unadjusted Function Point count can be calculated using the following:

Function Type	Complexity	Complexity Totals	Function Type Totals
ILF	____ Low	X 7	= ________
	____ Average	X 10	= ________
	____ High	X 15	= ________
		Sum	= ________

For example:

Assume that an application has 9 Internal Logical Files with complexities of: 3 Simple, 3 Average and 3 High. In this case, the Internal Logical File contribution to the application's Unadjusted Function Point count would be 96.

Function Type	Complexity	Complexity Totals	Function Type Totals
ILF	3 Low	X 7	= 21
	3 Average	X 10	= 30
	3 High	X 15	= 45
		Sum	= 96

External Interface Files. Each identified External Interface File is assigned a complexity based on the number of Record Element Types (RET) and Data Element Types (DET) associated.

Record Element Type Identification

Record Element Types (RET) are sub groupings of External Interface Files based on the logical/user view of the data. The Data Analysis equivalent to such logical groupings are Data Entities on a data flow diagram. External Interface Files which cannot be sub categorized are considered to have one RET.

One physical interface file can, when viewed logically, correspond to a number of External Interface Files. Additionally, multiple RETs can be associated to each External Interface File identified.

A method for identifying External Interface File RETs from one physical file is to look at the record types on the file. Exclude header and trailer records, unless specifically requested for audit purposes, as well as those record types required due to physical space limitations. Each unique record type corresponds to a RET.

Data Element Type Identification

An External Interface File's Data Element Types (DET) are user recognizable, non-recursive fields residing on the External Interface File. Each field on the External Interface File is a DET with the following exceptions:

- Fields should be viewed at the user recognizable level. For example, an account number or date which is physically stored in multiple fields should be counted as one DET.
- Fields which because of technology or implementation techniques appear more than once in an Internal Logical File should be counted only once. For example, if an External Interface File is comprised of more than one record type in a file, the record types used to identify the records would be counted only once.
- Repeating fields which are identical in format and exist so that multiple occurrences of a data value can occur, are counted only once. For example, an External Interface File containing 12 monthly budget amount fields and an annual budget amount field would be credited with 2 DETs, a DET for the monthly budget amount fields and a DET for the annual budget amount field.

Complexity assignments for External Internal Files (EIFs) is based on the following matrix:

	1 to 19 DET	20 to 50 DET	51 or more DET
1 RET	L	L	A
2 to 5 RET	L	A	H
6 or more RET	A	H	H

Legend
RET = Record Element Type
DET = Data Element Type (field)

Complexity
L = Low
A = Average
H = High

Use the following table to translate an External Interface File's complexity to Unadjusted Function Points.

Complexity Rating	Unadjusted Function Points
L (ow)	5
A (verage)	7
H (igh)	10

For example:

To calculate the Unadjusted Function Point count for an External Interface File having 23 DETs and 3 RET, you would use the EIF Complexity Matrix and find that the EIF is of Average Complexity. Secondly, using the EIF Unadjusted Function Point Table, transcribe the Average Complexity Rating to an Unadjusted Function Point value of 7.

An application's External Interface Files contribution to the total Unadjusted Function Point count can be calculated using the following:

Function Type	Complexity	Complexity Totals	Function Type Totals
EIF	_____ Low	X 5	= ________
	_____ Average	X 7	= ________
	_____ High	X 10	= ________
		Sum =	________

For example:

Assume that an application has 9 External Interface Files with complexities of: 3 Simple, 3 Average and 3 High. In this case, the External Interface File contribution to the application's Unadjusted Function Point count would be 66.

Function	Type Complexity	Complexity Totals	Function Type Totals
EIF	3 Low	X 5	= 15
	3 Average	X 7	= 21
	3 High	X 10	= 30
		Sum	= 66

External Inputs. Each identified External Input is assigned a complexity based on the number of File Type Referenced (FTR) and Data Element Types (DET).

File Type Referenced Identification

A File Type Referenced (FTR) should be counted for each Internal Logical File maintained and each External Interface File referenced during the processing of the External Input.

Data Element Type Identification

The Data Element Type (DET) count is the maximum number of user recognizable, non-recursive fields which are maintained on an Internal Logical File by the External Input.

Each field maintainable on an Internal Logical File by the External Input is a DET with the following exceptions:

- Fields should be viewed at the user recognizable level. For example, an account number or date which is physically stored in multiple fields should be counted as one DET.
- Fields which because of technology or implementation techniques appear more than once in an Internal Logical File should be counted only once. For example, if an Internal Logical File is comprised of more than one table in a relational DBMS, the keys used to relate the tables would be counted only once.

Additional DETs are credited to the External Input for the following capabilities:

- Command line(s), PF/action key(s) which provide the capability to specify the action to be taken by the External Input.
- One additional DET per External Input, NOT per action. Fields which are not entered by the user, but through the External Input are maintained on an Internal Logical File should be counted. For example, system generated date and time stamps which are maintained on an Internal Logical File for audit purposes.

Complexity assignments for External Inputs (EIs) is based on the following matrix:

	1 to 4 DET	5 to 15 DET	16 or more DET
0 to 1 FTR	L	L	A
2 FTR	L	A	H
3 or more FTR	A	H	H

Legend
FTR = File Type Referenced
DET = Data Element Type (field)

Complexity
L = Low
A = Average
H = High

Use the following table to translate an External Input's complexity to Unadjusted Function Points.

Complexity Rating	Unadjusted Function Points
L (ow)	3
A (verage)	4
H (igh)	6

For example:

To calculate the Unadjusted Function Point count for an External Input having 10 DETs and 2 FTR, you would use the EI Complexity Matrix and find that the EI is of Average Complexity. Secondly, using the EI Unadjusted Function Point Table, transcribe the Average Complexity Rating to an Unadjusted Function Point value of 4.

An application's External Inputs contribution to the total Unadjusted Function Point count can be calculated using the following:

Function Type	Complexity	Complexity Totals	Function Type Totals
EI	____ Low	X 3	= ________
	____ Average	X 4	= ________
	____ High	X 6	= ________
		Sum	= ________

For example:

Assume that an application has 9 External Inputs with complexities of: 3 Simple, 3 Average and 3 High. In this case, the External Input contribution to the application's Unadjusted Function Point count would be 39.

Function Type	Complexity	Complexity Totals	Function Type Totals
EI	3 Low	X 3	= 9
	3 Average	X 4	= 12
	3 High	X 6	= 18
		Sum	= 39

External Outputs. Each identified External Output is assigned a complexity based on the number of File Type Referenced (FTR) and Data Element Types (DET).

File Type Referenced Identification

A File Type Referenced (FTR) should be counted for each Internal Logical File and External Interface File referenced during the processing of the External Output.

Data Element Type Identification

A Data Element Type (DET) should be counted for each user recognizable, non-recursive field which appears on the External Output.

Each field appearing on the External Output is a DET with the following exceptions:

- Fields should be viewed at the user recognizable level. For example, an account number or date which is physically stored in multiple fields should be counted as one DET.
- Do not count literals as DETs.
- Do not count paging variables or system generated time/date stamps.

Additional DETs are credited to the External Output for the following:

- Count additional DETs for each summary or totalizer field appearing on the External Output.
- Error/confirmation messages: Count a DET for each distinct error/confirmation message available for display by the External Output.

Complexity assignment for External Outputs (EOs) is based on the following matrix:

	1 to 5 DET	6 to 19 DET	20 or more DET
0 to 1 FTR	L	L	A
2 to 3 FTR	L	A	H
4 or more FTR	A	H	H

Legend
FTR = File Type Referenced
DET = Data Element Type (field)

Complexity
L = Low
A = Average
H = High

Use the following table to translate an External Output's complexity to Unadjusted Function Points.

Complexity Rating	Unadjusted Function Points
L (ow)	4
A (verage)	5
H (igh)	7

For example:

To calculate the Unadjusted Function Point count for an External Output having 15 DETs and 2 FTR, you would use the EO Complexity Matrix and find that the EO is of Average Complexity. Secondly, using the EO Unadjusted Function Point Table, transcribe the Average Complexity Rating to an Unadjusted Function Point value of 5.

An application's External Outputs contribution to the total Unadjusted Function Point count can be calculated using the following:

Function Type	Complexity	Complexity Totals	Function Type Totals
EO	_____ Low	X 4	= _________
	_____ Average	X 5	= _________
	_____ High	X 7	= _________
		Sum	= _________

For example:

Assume that an application has 9 External Outputs with complexities of: 3 Simple, 3 Average and 3 High. In this case, the External Output contribution to the application's Unadjusted Function Point count would be 48.

Function Type	Complexity	Complexity Totals	Function Type Totals
EO	3 Low	X 4	= 12
	3 Average	X 5	= 15
	3 High	X 7	= 21
		Sum	= 48

<u>External Inquiries.</u>

Each identified External Inquiry is assigned a complexity based on the number of File Type Referenced (FTR) and Data Element Types (DET) on both the input and output sides of the inquiry.

File Type Referenced Identification - Input and Output sides

A File Type Referenced (FTR) should be counted for each Internal Logical File and External Interface File referenced during the processing of the External Inquiry.

Data Element Type Identification - Input side

A Data Element Type (DET) should be counted for those fields entered which specify the External Inquiry to be executed or specify data selection criteria.

Data Element Type Identification - Output side

A Data Element Type (DET) should be counted for each user recognizable, non-recursive field which appears on the output side of the External Inquiry.

Each field appearing on the External Output is a DET with the following exceptions:

- Fields should be viewed at the user recognizable level. For example, an account number or date which is physically stored in multiple fields should be counted as one DET. Fields which because of technology or implementation techniques appear more than once in an Internal Logical File should be counted only once.
- Do not count literals as DETs.
- Do not count paging variables or system generated time/date stamps.

Additional DETs are credited to the External Output for the following:

- Count additional DETs for each summary or totalizer field appearing on the External Output.
- Help messages: There are three categories of Help, Full Screen Help, Field Sensitive Help and Help Subsystems. DET determination varies between each and is discussed below:
    - Full Screen Help: Credit a Low Complexity External Inquiry regardless of the number of FTRs or DETs involved.
    - Field Sensitive Help: Credit an External Inquiry having a complexity, using the input side, based on the number of fields which are field sensitive and the number of FTRs. Each field sensitive field corresponds to an DET.
    - Help Subsystems - Specific counting rules are not available at this time.

Complexity assignment for External Inquiries are based on the following matrices:

External Inquiry (EQ) Input Complexity Matrix

	1 to 4 DET	5 to 15 DET	16 or more DET
0 to 1 FTR	L	L	A
2 FTR	L	A	H
3 or more FTR	A	H	H

Legend
FTR = File Type Referenced
DET = Data Element Type (field)

Complexity
L = Low
A = Average
H = High

External Inquiry (EQ) Output Complexity Matrix

	1 to 5 DET	6 to 19 DET	20 or more DET
0 to 1 FTR	L	L	A
2 to 3 FTR	L	A	H
4 or more FTR	A	H	H

Legend
FTR = File Type Referenced
DET = Data Element Type (field)

Complexity
L = Low
A = Average
H = High

Use the following table to translate an External Inquiry's complexity to Unadjusted Function Points.

Complexity Rating	Unadjusted Function Points
L (ow)	3
A (verage)	4
H (igh)	6

For example:

To calculate the Unadjusted Function Point count for an External Inquiry having for input 2 DETs and 1 FTR, and for output 10 DETs and 2 FTRs, you would use the EQ Input Complexity Matrix and find that the input is a Low Complexity. Secondly, using the EQ Output Complexity Matrix and find that the output is an Average Complexity.

Next, comparing the "Low" Complexity Rating from the input side with the "Average" Complexity Rating from the output side, select the higher of the two as the External Inquiry's Complexity Rating.

Using the EQ Unadjusted Function Point Table, transcribe the Average Complexity Rating to an Unadjusted Function Point value of 4.

An application's External Inquiries contribution to the total Unadjusted Function Point count can be calculated using the following:

Function Type	Complexity	Complexity Totals	Function Type Totals
EQ	____ Low	X 3	= ________
	____ Average	X 4	= ________
	____ High	X 6	= ________
		Sum	= ________

For example:

Assume that an application has 9 External Inquiries with complexities of: 3 Simple, 3 Average and 3 High. In this case, the External Inquiry contribution to the application's Unadjusted Function Point count would be 48.

Function Type	Complexity	Complexity Totals	Function Type Totals
EQ	3 Low	X 3	= 9
	3 Average	X 4	= 12
	3 High	X 6	= 18
		Sum	= 39

B. Value Adjustment Factor Calculation

The Value Adjustment Factor is based on 14 General System Characteristics which rate the overall complexity of the application. The 14 General System Characteristics are summarized into the Value Adjustment Factor. When applied, the Value Adjustment Factor adjusts the Unadjusted Function Point count +/- 35 %, producing the final Function Point count.

1) Evaluate the 14 General System Characteristics (GSC) on a scale from 0 to 5 producing a Degree of Influence (DI) for each of the General System Characteristic questions.
2) Sum the 14 DIs to produce the Total Degree of Influence (TDI).
3) Insert the TDI into the following equation to produce the Value Adjustment Factor (VAF):

   $(TDI * 0.01) + 0.65 = VAF$

   Where:
   TDI = the sum of the 14 Degrees of Influence
   VAF = the Value Adjustment Factor

The following table can be used to facilitate the calculation of the VAF:

General System Characteristics (GSC)	Degree of Influence (DI) 0 - 5
1. Data Communication	________
2. Distributed Processing	________
3. Performance	________
4. Heavily use configuration	________
5. Transaction rates	________
6. On-line data entry	________
7. Design for end user efficiency	________
8. On-line update	________
9. Complex processing	________
10. Usable in other applications	________
11. Installation ease	________
12. Operational ease	________
13. Multiple sites	________
14. Facilitate change	________
Total Degree of Influence (TDI)	________
Value Adjustment Factor (VAF) (________ X 0.01) + 0.65 = VAF = (TDI * 0.01) + 0.65	________

C. Application Function Point Calculation

To produce the initial Application Function Point count, complete the following formula:

UFP * VAF = AFP

Where:
UFP = the Unadjusted Function Point count
VAF = the Value Adjustment Factor
AFP = the initial Application Function Point count

To produce the revised Application Function Point count, complete the following formula:

((UFPB + ADD + CHGA) - (CHGB + DEL)) * VAFA = AFP

Where:
UFPB = the Application's Unadjusted Function Point prior to the enhancement project.
ADD = the Unadjusted Function Point count of those functions which were added by the enhancement project. CHGA = the Unadjusted Function Point count of those functions which were modified by the enhancement project. This number reflect the functions AFTER the modifications.
CHGB = the Unadjusted Function Point count of those functions which were modified by the enhancement project. This number reflect the functions BEFORE the modifications.
DEL = the Unadjusted Function Point count of those functions which were deleted by the enhancement project.
VAFA = the Value Adjustment Factor of the application after the enhancement project.
AFP = the revised Application Function Point count

D. Development (Project) Function Point calculation

To produce the Development (Project) Function Point count, complete the following formula:

UFP * VAF = DFP

Where:
UFP = the Unadjusted Function Point count
VAF = the Value Adjustment Factor
DFP = the Development (Project) Function Point count

Appendix F - Methodology II, the Entity-Relationship Methodology

FUNCTION POINTS COUNTING METHODOLOGY II

For Use By Raters C and D

I. FUNCTION POINT ANALYSIS OVERVIEW

A. Objectives of Function Point Analysis

Function Points measure software by quantifying the functionality provided external to itself, based primarily on logical design. With this in mind, the objectives of Function Point counting include:

- o provide a normalization factor for software comparison;
- o provide a sizing metric allowing for productivity analysis;
- o measure independently of technology used for implementation;
- o provide a vehicle for software estimation;
- o measure what the user requested and received;

In addition to meeting the above objectives, the process of counting function points should be:

- o simple enough to minimize the overhead of the measurement process;
- o simple yet concise, to allow for consistency over time, projects and practitioners.

B. Summary of Function Point Counting

The Function Point metric measures an application based on two areas of evaluation. The first results in the Unadjusted Function Point count and reflects the functionality provided to the user by the application. The second area of evaluation, which produces the Value Adjustment Factor (VAF), evaluates the overall complexity of the application.

The Function Point metric measures the application in terms of WHAT is delivered not HOW it is delivered. Only user-requested and visible components are counted. These components called Function Types and are categorized as either Data or Transactional.

Function Types:

Data:

Internal Logical Files (ILF) - internally maintained logical group of data.
External Interface Files (EIF) - externally maintained logical group of data.

Transactional:

External Inputs (EI) - maintains internally stored data.
External Outputs (EO) - standardized data output.
External Inquiries (EQ) - on-line combination of input (request) and output (retrieval).

Each Function Type is further categorized based on its complexity.

- Low
- Average
- High

Complexity rating criteria vary between Function Type, and is discussed in Section II, Counting Rules.

Function Point values, ranging from 3 to 5 depending on the Function Type and complexity rating, are assigned and totaled producing the Unadjusted Function Point Count.

The resulting number is then weighted by the Value Adjustment Factor (VAF) to produce the Adjusted Function Point Count. The VAF is comprised of 14 General System Characteristic (GSC) questions. The GSC questions assess overall system requirements.
The General System Characteristics are:

1. Data Communication
2. Distributed function
3. Performance
4. Heavily use configuration
5. Transaction rates
6. On-line data entry
7. Design for end user efficiency
8. On-line update
9. Complex processing
10. Usable in other applications
11. Installation ease
12. Operational ease
13. Multiple sites
14. Facilitate change

The questions are answered using Degrees of Influence (DI) on a scale of 0 to 5.

0 Not Present, or no influence
1 Incidental influence
2 Moderate influence
3 Average influence
4 Significant influence
5 Strong influence throughout

C. Types of Function Point Counts

Function Point counts can be associated to either projects or applications. There are three types of function point counts:

1. An Application Function Point Count reflects the installed functionality for an application. Also called a Baseline or Installed Function Point Count, it reflects the installed functionality of the application.

2. Development (Project) Function Point Count - Function Point count associated with the initial installation of new software. This count reflects the installing project's user functionality.

3. Enhancement (Project) Function Point Count - Function Point count associated with the enhancement of existing software. An Enhancement Function Point Count reflects those modifications to an existing application which add, change or delete user functionality.

D. Application of Logical Models

This methodology's definition of function point counting is based on the use of logical models as the basis of the counting process. The two primary models which are to be used are the "Data-Entity-Relationship" model and the "Data Flow Diagram." These two model types come in a variety of forms, but generally have the same characteristics related to Function Point counting irrespective of their form. The following applies to these two models as they are applied in the balance of this document.

Data Entity Relationship Model (DER). This model typically shows the relationships between the various data entities which are used in a particular system. It typically contains "Data Entities" and "Relationships", as the objects of interest to the user or the systems analyst. In the use of the DER model, we standardize on the use of the "Third Normal Form" of the model, which eliminates repeating groups of data, and functional and transitive relationships. No further discussion of data modeling is provided in this document, as the subject is well covered in many other texts and references.
Data Entity Relationship models will be used to identify Internal Entities (corresponding to Logical Internal Files) and External Entities (corresponding to Logical External Interfaces).

Data Flow Diagrams (DFD). These models typically show the flow of data through a particular system. They show the data entering from the user or other source, the data entities which are used, and the destination of the information out of the system. The boundaries of the system are generally clearly identified, as are the processes which are used. This model is frequently called a "Process" model. The level of detail of this model which is useful is the level which identifies a single (or small number) of individual business transactions. These transactions are a result of the decomposition of the higher level data flows typically at the system level, and then at the function and sub-function level.

Data Flow Diagrams will be used to identify the three types of transactions which are counted in Function Point Analysis (External Inputs, External Outputs and Inquiries).
Logical Internal Files

E. Boundaries

It is useful to distinguish between establishing boundaries when counting the function points on an existing system versus on a project. The following guidelines are organized to reflect the guidance which applies to counting points on established systems and those in projects.

System

- o Look at the application from the POINT OF VIEW OF THE USER, what the user can touch and feel. Use the system external specs or get a systems flow chart and draw a boundary around it to highlight what is internal to the application vs. what is external.
- o An application is an implemented system, the maintenance of which is managed as a unit. The application boundary encloses this system. All functions within the boundary are counted and recorded together.
- o When a development project is implemented as incremental functionality for an existing application currently being maintained, the function points are added to the application in order to measure future maintenance. An exception are the interfaces between the newly implemented functions and the existing application. These interfaces are eliminated (the interfaces are now within the expanded application boundary).

Project

- o A development project boundary encloses each exchange of data between user and for other projects/systems is used to identify project's function.
- o List all of the following that are made available to the USER through the design, programming, testing, or maintenance efforts of the project team:

  - Inputs from the user
  - Outputs to the user
  - Logical data groups as viewed by the user
  - Interfaces to/from other systems
  - Inquiries

  selected, modified, integrated, tested, and installed (i.e. provided for beneficial use to the user through the efforts of the project team).

Counting Enhancements

- o For enhancements, only count those parts of the system for which you made a change to the format or processing. Count them as they exist after the enhancement. Do not include the "before picture". Count user functions that were deleted, because they will be ADDED to the total function points.
- o An enhancement can be required that does not change any inputs, outputs, or files but involves a great deal of time, eg. the internal logic of a pricing system must be changed to derive the product price differently.

Counting Packages

- o Package systems may supply function the user did not ask for. Only count the part that is actually used. Also, count function points for everything the user accepted.
- o A project to buy a package and make changes to it is 100% software acquisition. If some changes have to be made to an existing system to make the package fit in, then there is a split percentage and part would be enhancement, to other systems and part would be software acquisition.
- o Once a package is in place, projects to install updates are enhancements even if the new versions have to be paid for.

F. Users

- o Ensure that you include Operations and the Audit department as your users. This means that you include any audit logs or recovery files that are not "work files" i.e. exact copies of files already counted for some other reason.

G. Complexity

- o The complexity matrices are designed to promote uniformity; however, do not be afraid to use your judgement - if you think an output is simple, then classify it that way!
- o List a function even if you do not know how to classify it. When in doubt, classify as average. Don't assume that the documentation shown is complete or correct. Oversight of functions (particularly in the early stages of development) is the main cause of inaccuracy.

H. Security

- o Sign on security is usually a constant for most applications. However, if additional application security is required count it on the appropriate screens.
- o Logon procedures which necessitate security checks in a conversational mode are considered as one input security function. An unsecured logon procedure would have no function since it is considered part of the basic function being referenced.

I. Operating Systems & Utilities

- o These do not generally provide direct user visible business function. Thus they are not counted for Function Point purposes. The project time spent an operating system and utilities is part of the effort required to deliver function to the users and should be counted for time accounting purposes.

K. Report Writers

- o A report writer or inquiry facility that is made available for users to define ad hoc reports or inquiries should be decomposed into its components of Logical Data Structures and User Interfaces using the existing definitions and current practices.

II. COUNTING RULES

General

Function Points are calculated based on user-requested, visible components of an application. These components are called Function Types and are categorized as either Data or Transactional.
Data Function Types evaluate the functionality provided the user for internal and external data requirements.

Internal Logical Files (ILF) reside internal to an application's boundary and reflect data storage functionality provided to the user. Internal Logical Files must be maintained and utilized by the application.

External Interface Files (EIF) reside external to an application's boundary and reflect the functionality provided by the application through the use of data maintained by other applications.
While both Internal Logical Files and External Interface Files contain the word "file" in their title, they are not files in the traditional DP sense of the word. In this case, file refers to a logically related group of data and not the physical implementation.

Transactional Function Types evaluate the functionality provided the user for the processing requirements of an application.

External Inputs (EI) reflect the functionality provided the user for the receipt of and maintenance of data on Internal Logical Files.

External Outputs (EO) reflect the functionality provided the user for output generated by the application.

External Inquiries (EQ) reflect the functionality provided the user for on-line queries of Internal Logical Files or External Interface Files.

A. Internal Logical Files

Definition. Internal entity types are counted as Albrecht's internal file types. An entity-type is internal if the application built by the measured project allows users to create, delete, modify and/or read an implementation of the entity-type. The users must have asked for this facility and be aware of it.

All attributes of the entity-type, elements that are not foreign keys, are counted. We also count the number of relation types that the entity-type has.

The complexity is determined by counting the number of elements and the number of relationships. The points assignment rule for internal entities is:

Number of Data Attributes Types in the Entity

	1 to 19 ELE	20 to 50 ELE	51 or more ELE
1 REL	L	L	A
2 to 5 REL	L	A	H
6 or more REL	A	H	H

Legend
REL = Relationships or other Entity Types
ELE = Data Element

Complexity
L = Low
A = Average
H = High

Use the following table to translate a Internal Logical File's complexity to Unadjusted Function Points.

Complexity Rating	Unadjusted Function Points
L (ow)	7
A (verage)	10
H (igh)	15

Guidance

- o View data from the user's standpoint - concentrate on the system's external, conceptual design instead of its internal, physical design. Ask the question - how would the user file the data if this was a manual system?
- o Count the following as 1 logical internal file type:
  - logical entry of data from user viewpoint
  - logical internal files generated or maintained by the application
- o Count the number of relationships in which the logical internal file participates. Relationships are between key data in the data group and other data groups.
- o A logical entity of data from the user viewpoint is one logical internal file ie. a block of interrelated data derived from user requirements.

- o If a file is not part of a user's requirements, don't count it.
- o Count the elements at the generally recognized level (eg. start date, NOT start month, start day, etc.).
- o When evaluating a logical file, Internal or External, only count the elements and relationships required by the application.
- o Function Point Analysis distinguishes between two types of files: files with temporary transactions and files with logical records of permanent data. Even though both kinds of data are physically stored in files, only the latter is counted as being a "logical" file.
- o Stores of persistent or permanent data are viewed as logical files. The following terms are often used to describe logical files: "persistent data", "status", "record", "master file", "updated records", "reference files" and "historical data". When used and maintained within the application, classify them as logical internal files. When shared between applications, they are classified as external interface files.
- o Transactions are considered to be events that trigger changes or updates to logical internal files; they are NOT classified as files themselves. The following terms are used to describe transactions: "event", "transient data", "stimulus", "trigger", and "update data". A transaction file can be classified as an external input type if it is read to process data in a logical internal file.
- o Files consisting of transactions (such as a tape with MICR item images) count as input types or output types or possibly as an external interface, but not as a file type.

<u>Examples</u>

- o logical data base records
- o logical groups of project data accessible to user

<u>Data Modeling (Entity Relationship Modeling)</u>

- o Entities updated by application are counted as logical internal files.
- o Complexity is based on the number of relationships in which the entity participates as well as the number of DET's.
- o When considering an Entity-Relationship chart, be sure to consider the real needs of the application. For instance, frequently attributes required are attributes of the relationship rather than the entities, thus requiring a concatenated key to satisfy the requirement. The related entities may or may not be required as separate USER VIEWS.

B. External Interfaces

Definition. Entity-types which are used by the application but not maintained by it are counted as external interfaces. If the user does not have the capability to maintain it, then it is not an internal entity-type.

These may be implemented as parts of a shared data base, ordinary files, or look-up tables "copied" into programs.

The counting of attributes and entity relationship-types is the same as for internal entity-types.
The points assignment rule for external entities is:

Number of Data Attributes Types in the Entity

	1 to 19 ELE	20 to 50 ELE	51 or more ELE
1 REL	L	L	A
2 to 5 REL	L	A	H
6 or more REL	A	H	H

Legend
REL = Number of Relationships to other Entity-Types
ELE = Data Element

Complexity
L = Low
A = Average
H = High

Use the following table to translate an External Interface's complexity to Unadjusted Function Points.

Complexity Rating	Unadjusted Function Points
L (ow)	5
A (verage)	7
H (igh)	10

Guidance.

- o There is a distinct difference between external interface files and transactions. Transactions are events which affect logical files. Applications can interface with each other through transactions or external interface files. If applications provide data through transactions; inputs, outputs, and/or inquiries are counted. The key to an external interface file is that an application must be able to access the data directly without the aid of another application.
- o Do not identify input, output, or inquiry function types for an external interface file, as they are transactions.

- o If a master file is used by a system but not maintained, it is an external interface. A file containing transactions from another system to update files in this system is considered an external input and not an external interface.

Examples.

- o POTENTIAL EXTERNAL INTERFACE FILES
  - logical file shared by applications provided it is used as a REFERENCE file, and not as a TRANSACTION file
  - logical file from another application used for reference only
- o Count the following as 1 external interface type:
  - file of records from another application
  - file of records to multiple other applications (multiple distribution is a level of information processing function consideration)
  - data base shared from other applications
- o Exclusions from the count:
  - a logical internal file used by another application as a transaction (Input) file. This is counted as an external output.
  - a logical internal file coming from another application that is used as a transaction (external input) file.

C. External Inputs

Definition. An input causes some change in the data stored in the application. It must come from the user. It may come directly, as an input screen, or indirectly, as a file of update transactions from another application.

The logic required for adding a new entity occurrence to the data base is different from that required for update or delete. Therefore, one set of attributes used with these functions usually counts as three inputs (add, change, delete).

The number of elements used in an input and the number of entity-types in which those elements appear are counted.

The points assignment rule for inputs is:

Number of Data Attributes Types in the Entity

	1 to 4 ELE	5 to 15 ELE	16 or more ELE
1 REL	L	L	A
2 REL	L	A	H
3 or more FTR	A	H	H

Legend
REL = Number of Relationships or other Entity Types
ELE = Data Element

Complexity
L = Low
A = Average
H = High

Use the following table to translate an External Input's complexity to Unadjusted Function Points.

Complexity Rating	Unadjusted Function Points
L (ow)	3
A (verage)	4
H (igh)	6

Guidance.

- o Count input types that enter from other applications as well as from the end user.
- o Do not count inquiry transactions. These are counted as inquiries.
- o The count for file types referenced (FTR) is simple after the logical internal files have been defined. The data elements (DET's) must be identified for each logical internal file. The DET's of the input function type may correspond to some or all of the DET's of a logical internal file. The DET's belonging to the logical internal file are not counted as input DET's; only the DET's that are directly impacted by the input are counted.

- If an inquiry is made to obtain information which results in data being entered to change a record, it is counted as an external inquiry and an input.
- A file containing transactions from another system to update files in this system is considered an external input and not an external interface.

D. External Outputs

Definition. An output is some flow out of the application that happens as a regular part of the processing independently of conditions set in the data. A monthly report is produced every month regardless of what has happened, and is counted as an output. Inquiries are triggered, directly or indirectly by the user, and are not counted as outputs.

The same sets of attributes presented with different technology are not separate output-types (e.g.. a report on paper and on microfilm are not separate output-types).

All data elements that are actually output are counted. Derived data elements such as totals are counted. Each different level of summarization is counts as a data element.

Repeats of the same data, like lines of a report are not counted. Preprinted formats are not counted.

All foreign key, identifiers and regular attributes are counted.

The point assignment rule for outputs is:

Data Elements Types in the Output	1 to 5 ELE	6 to 19 ELE	20 or more ELE
1 Entity-Type	L	L	A
2 to 3 Entity-Type	L	A	H
4 or Entity-Type . FTR	A	H	H

Legend
Entity-Type = Entities Referenced in the Output
ELE = Data Element

Complexity
L = Low
A = Average
H = High

Use the following table to translate an External Output's complexity to Unadjusted Function Points.

Complexity Rating	Unadjusted Function Points
L (ow)	4
A (verage)	5
H (igh)	7

Guidance.

- o Include data used by another application if the data is used as a transaction by that application (eg. if it is used to update a file).

- o Don't confuse outputs with the response side of an external inquiry type. If data is displayed in the exact form that it was entered into the system, it is an external inquiry type. If the raw data has ever been manipulated within the system, it is considered a true output.
- o DET's which are a result of calculations or coding conversions are also counted. These DET's are more difficult to identify because they are not necessarily displayed to the user or contained in logical internal files.
- o Note that any user-maintained table or file may require at least the following additional functions:
  - one input
  - one output
  - one inquiry
- o A file created by one process and passed to another for printing is normally counted as an external output.
- o A second report is produced which duplicates another, except it eliminates some data. Since it satisfies a different user function, it is considered to be another external output.

E. External Inquiry

Definition. An inquiry is an output of information in response to some user action. No change to the data base takes place. This may be on a terminal, a report or other forms. The inquiry may be in direct response to a user request for information.

Inquiries have an input part, the trigger, and an output part, the response. The points calculated for the most complex part are counted; this is almost always the output part.

All entity-types which are required to be accessed are counted.

If data-elements of an entity-type are in the output-part, they may be counted. The points assignment rule for the input and output of inquiries is:

Data Elements Types in the QT-Input	1 to 4 ELE	5 to 15 ELE	16 or more ELE
1 Entity-Type	L	L	A
2 Entity-Type	L	A	H
3 or more Entity-Type . FTR	A	H	H

Legend
Entity-Type = Entities Referenced in the Inquiry
ELE = Data Element

Complexity
L = Low
A = Average
H = High

Use the following table to translate an Inquiry's complexity to Unadjusted Function Points.

Complexity Rating	Unadjusted Function Points
L (ow)	3
A (verage)	4
H (igh)	6

Data Elements Types in the QT-Output	1 to 5 ELE	6 to 19 ELE	20 or more ELE
1 Entity-Type	L	L	A
2 to 3 Entity-Type	L	A	H
4 or Entity-Type . FTR	A	H	H

Legend
Entity-Type = Entities Referenced in the Output
ELE = Data Element

Complexity
L = Low
A = Average
H = High

Use the following table to translate an Inquiry's complexity to Unadjusted Function Points.

Complexity Rating	Unadjusted Function Points
L (ow)	4
A (verage)	5
H (igh)	7

Guidance.

- o Do not count inquiries as external inputs or outputs.
- o Overnight or delayed inquires which create a file for intermediate storage after queries can be counted as separate inputs and outputs and files only if the user recognizes the file as a separate logical data entity.
- o An inquiry and an input may be juxtaposed, sharing a common physical screen which serves as the output side of the inquiry and the input screen. Count these as separate logical transactions.
- o The first issue to be resolved is whether the user identifies the transaction as are or several logical transactions. If the interim screens are viewed as delivering separable function, and can be viewed as end points of the inquiry, they should be counted as multiple inquiries. Otherwise, the linked screens are one inquiry with the last screen as the inquiry - output and the preceding screens as the inquiry - input.

Examples.

- o Count the following as indicated:
  - inquiry followed by an update input count as 1 inquiry and 1 input

POTENTIAL EXTERNAL INQUIRIES.

- user inquiry with no file update
- help messages and screens

E. Additional Guidance

Menus recommendation. Menus do not appear on logical models of systems. They do not directly provide user function, but ease access to the function. Menus are not counted as:

- inputs
- outputs
- inquiries or
- files

Existence of extensive menus, as part of a user interface and may change the General Computing Adjustments.

Error Messages recommendation. Error messages are part of input, output, and inquiry transactions and are counted as DET's in relation to a specific transaction.

Backup Files recommendation. Backup files are duplicates of internal physical files. They are separate physical representations of internal "logical entities." These separate physical representations do not appear on the logical data model, thus they are not counted as either files or outputs.

Certain applications recognize "Archived" data as having a separate and distinct identity from the active data. In these cases the archived data can be recognized as separate entities.

External Interface Files Guidelines recommendation. In logical data modeling, all data is viewed as being accessible by other applications. Thus virtually every internal entity could be "double counted" as an external entity. In the interest of conservative counting, a data entity should be counted as either internal or external relative to the project/system being counted.

Update/Inquiry Screens and Implied Inquiries recommendation. An inquiry and an input are logically separate transactions. In practice they may be juxtaposed, sharing a common physical screen which serves as the output side of the inquiry and the input screen. Count these as separate logical transactions.

Multiple Screens for One Inquiry recommendation. The first issue to be resolved is whether the user identifies the transaction as are or several logical transactions. If the interim screens are viewed as delivering separable function, and can be viewed as end points of the inquiry, they should be counted as multiple inquiries. Otherwise, the linked screens are one inquiry with the last screen as the inquiry - output and the preceding screens as the inquiry - input.

G. Physical Guidelines

Hierarchical Databases.

- All segments below the root segment of a hierarchical data base (eg. basic IMS) logically have concatenated keys. The USER VIEW consists of a complete path (including the root segment) of such a data base.
- Truncations of a complete path of a hierarchical data base do not count as additional user views. They are merely sub-sets of the more complete user view.
- When any segments are accessed by logical paths of secondary indexes the resultant USER VIEWS are considered to be separate structures. Count only the USER VIEWS that are based on USER REQUIREMENTS and not those that are there because the implementation made them available.
- When hierarchical data bases have segments for technological reasons, such a segment (no-key segments, fragmentation of a field into multiple levels for performance considerations, etc.) should be considered as part of the parent segment with the keys concatenated.
- When two or more physical files have the same key, they are considered to be a single USER VIEW.
- Sequence of data provides unique USER VIEWS. A concatenated key consisting of FIELD-A and FIELD-B in that order provides a different USER VIEW than FIELD-B and FIELD-A in that order. User access must be provided for the data structures in order to produce reports does not count. A file sorted to a different sequence and made available for user inquiries does not count.
- If a single segment of a complex data base is accessed for all transactions (eg. a name and address), it is functionally the same as a flat file. If multiple segments are accessed, the single segment is counted as any other segment.

Relational Databases.

- When considering what USER VIEWS are present in a relational data base, consider JOINS only where they are necessary to satisfy a USER REQUIREMENT.

III. GENERAL SYSTEMS CHARACTERISTICS

Each General System Characteristic (GSC) must be evaluated in terms of its Degree of Influence (DI) on a scale of 0 to 5. The descriptions listed under "Score as:" are meant to be guides in determining the Degree of Influence. If none of the guideline descriptions fit the application exactly, a judgement must be made about which Degree of Influence most closely applies to the application.

1. Data Communications

The DATA and CONTROL information used in the application are sent or received over communication facilities. Terminals connected locally to the control unit are considered to use communication facilities. Score as:

0 - Application is pure batch processing or a stand alone PC.

Remote data entry/printing

1 - Application is batch but has remote data entry or remote printing.

2 - Application is batch but has remote data entry and remote printing.

Interactive teleprocessing (TP)

3 - On-line data collection or TP front end to a batch process or query system.

4 - More than a front-end, but the application supports only one type of TP communications protocol.

5 - More than a front-end, but the application supports more than one type of TP communications protocol.

2. Distributed Data Processing

Distributed data or processing functions are a characteristic of the application within the application boundary. Score as:

0 - Application does not aid the transfer of data or processing function between components of the system.

1 - Application prepares data for end user processing on another component of the system (e.g. PC spreadsheet, PC DBMS, etc.).

2 - Data is prepared for transfer, is transferred, and is processed on another component of the system (not for end user processing).

3 - Distributed processing and data transfer is on-line and in one direction, only.

4 - Distributed processing and data transfer is on-line and in both directions.

5 - Processing functions are dynamically performed on the most appropriate component of the system.

3. Performance

Application performance objectives, stated or approved by the user, in either response or throughput, influenced (or will influence) the design, development, installation and support of the application. Score as:

0 - No Special performance requirements were stated by the user.

1 - Performance and design requirements were stated and reviewed but no special actions were required.

2 - On-line response time is critical during peak hours. No special design for CPU utilization was required. Processing deadline is for the next business day.

3 - On-line response time is critical during all business hours. No special design for CPU utilization was required. Processing deadline interface systems is not constraining.

4 - Stated user performance requirements are stringent enough to require performance analysis tasks in the design phase.

5 - In addition, performance analysis tools were used in the design, development, and/or implementation phases to meet the stated user performance requirements.

4. Heavily Used Configuration

A heavily used operational configuration, requiring special design considerations, is a characteristic of the application (e.g., the user wants to run the application on existing or committed equipment that will be heavily used). Score as:

0 - No explicit or implicit operational restrictions.

1 - Operational restrictions do exist, but are less restrictive than a typical application. No special effort is needed to meet the restrictions.

2 - Some security or timing considerations.

3 - Specific processor requirements for a specific piece of the application.

4 - Stated operation restrictions require special constraints on the application in the central processor, or, a dedicated processor.

5 - In addition, there are special constraints on the application in the distributed components of the system.

5. Transaction Rate

The transaction rate is high and it influenced the design, development, installation, and support of the application. Score as:

0 - No peak transaction period anticipated.

1 - Monthly peak transaction period anticipated.

2 - Weekly peak transaction period anticipated.

3 - Daily peak transaction period anticipated.

4 - High transaction rates stated by the user in the application requirements or service level agreements are high enough to require performance analysis tasks in the design phase.

5 - High transaction rates stated by the user in the application requirements or service level agreements are high enough to, in addition, require the use of performance analysis tools in the design, development, and/or installation phases.

6. On-Line Data Entry

On-line data entry and control functions are provided in the application. Score as:

0 - All transactions are processed in batch mode.

1 - 1% - 7% of transactions are interactive data entry.

2 - 8% - 15% of transactions are interactive data entry.

3 - 16% - 23% of transactions are interactive data entry.
4 - 24% - 30% of transactions are interactive data entry.
5 - Over 30% of transactions are interactive data entry.

7. End User Efficiency

The on-line functions provided emphasize a design for end user efficiency. Efficiency can be delivered in one or more of the following forms:

- menus
- on-line help/documentation
- scrolling
- remote printing (via on-line transactions)
- preassigned function keys
- submission of batch jobs from on-line transactions
- cursor selection of screen data
- heavy use of reverse video, highlighting, colors underlining, etc.
- hard copy user documentation of on-line transactions
- mouse interface

- pop-up windows
- as few screens as possible to accomplish a business function
- easy navigation between screens (e.g., through function keys)
- bi-lingual support (supports 2 languages; count as 4 items)
- multi-lingual support (supports more than 2 languages; count as 6 items)

Score as:

0 - none of the above

1 - 1 - 3 of the above.

2 - 4 - 5 of the above.

3 - 6 or more of the above, but there are no specific user requirements related to efficiency.

4 - 6 or more of the above, and, stated requirements for end-user efficiency are strong enough to require design tasks for human factors to be included (e.g., minimize key strokes, maximize defaults, use of templates, etc.).

5 - 6 or more of the above, and, stated requirements for end-user efficiency are strong enough to require use of special tools and processes in order to demonstrate that the objectives have been achieved.

8. On-Line Update

The application provides on-line update for the internal logical files. Score as:

0 - None.

1 - On-line update of 1 to 3 control files. Volume of updating is low and recovery is easy.

2 - On-line update of 4 or more control files. Volume of updating is low and recovery is easy.

3 - On-line update of major internal logical files.

4 - In addition, protection against data lost is essential and has been specially designed and programmed in the system.

5 - In addition, high volumes bring cost considerations into the recovery process. Highly automated recovery procedures with minimum of operator intervention.

9. Complex Processing

Complex processing is a characteristic of the application. Categories are:

- Sensitive control (e.g., special audit processing) and/or application specific security processing.
- Extensive logical processing.
- Extensive mathematical processing.
- Much exception processing resulting in incomplete transactions that must be processed again; e.g., incomplete ATM transactions caused by a line drop.
- Complex processing to handle all input/output possibilities; e.g., multi-media, device independence, etc.

Score as:

0 - None of the above.

1 - Any one of the above.

2 - Any two of the above.

3 - Any three of the above.

4 - Any four of the above.

5 - Any five of the above.

10. Reusability

The application, and the code in the application, have been specifically designed, developed, and supported to be usable in other applications. Score as:

0 - No reusable code

1 - Reusable code is used within the application.

2 - Less than 10% of the modules produced considered more than one users needs.

3 - More than 10% of the modules produced considered more than one users needs.

4 - The application was specifically packaged and/or documented to ease re-use, and application is customized by user at source code level.

5 - The application was specifically packaged and/or documented to ease re-use, and applications is customized for use by means of parameter input.

11. Installation Ease

Conversion and installation ease are characteristics of the application. A conversion and installation plan and/or conversion tools were provided, and were tested during the system test phase. Score as:

0 - No special considerations stated by user and no special set up required for installation.

1 - No special considerations stated by user BUT special set up required for installation.

2 - Conversion and installation requirements were stated by the user and, conversion and installation guides were provided, and tested. The impact of conversion on the project is not considered to be important.

3 - Conversion and installation requirements were stated by the user and, conversion and installation guides were provided, and tested. The impact of conversion on the project is considered to be important.

4 - In addition to (2), automated conversion and installation tools were provided and tested.

5 - In addition to (3), automated conversion and installation tools were provided and tested.

12. Operational Ease

Operational ease is a characteristic of the application. Effective start-up, back-up, and recovery procedures were provided, and were tested during the system test phase. The application minimizes the need for manual activities, such as tape mounts, paper handling, and direct on-location manual intervention. Score as:

0 - No special operational considerations other than the normal backup procedures, were stated by the user.

1 to 4 - Select which of the following items apply to the application. Each item has a point value of one, except as noted otherwise.

-Effective startup, backup, and recovery processes were provided but operator intervention is required.
-Effective startup, backup, and recovery processes were provided but operator intervention is required. (count as 2 items).
-The application minimizes the need for tape mounts.
-The application minimizes the need for paper handling.

5 - Application is designed for unattended operation. Unattended operation means NO OPERATOR INTERVENTION required to operate the system other than to start up or shut down the application. Automatic error recovery is a feature of the application.

13. Multiple Sites

The application has been specifically designed, developed, and supported to be installed at multiple sites for multiple organizations. Score as:

0 - No user requirement to consider the needs of more than one user site.

1 - Needs of multiple sites were considered in the design and the application is designed to operate only under identical hardware and software environments.

2 - Needs of multiple sites were considered in the design and the application is designed to operate only under similar hardware and/or software environments.

3 - Needs of multiple sites were considered in the design and the application is designed to operate under different hardware and/or software environments.

4 - Documentation and support plan are provided and tested to support the application at multiple sites and application is as described by (1) or (2).

5 - Documentation and support plan are provided and tested to support the application at multiple sites and application is as described by (3).

14. Facilitate Change

The application has been specifically designed, developed, and supported to facilitate change. Examples are:

- Flexible query capability is provided.
- Business control data is grouped in tables maintainable by the user.

Score as:

0 - No special user requirement to design the application to minimize or facilitate change.

1 to 5 - Select which of the following items apply to the application. Each item has a point value of one except as noted otherwise.

- Flexible query facility is provided which can handle simple query requests; e.g., ANDs or ORs logic applied to only one internal logical file.
- Flexible query facility is provided which can handle query requests of average complexity: e.g., ANDs or ORs logic applied to more than one internal logical file . (count as 2 items)
- Flexible query facility is provided which can handle complex query requests: e.g., AND/OR logic combinations on one or more internal logical files. (count as 3 items)

- Control data is kept in tables that are maintained by the user with on-line interactive processes but changes take effect only on the next business day.
- Control data is kept in tables that are maintained by the user with on-line interactive processes and the changes take effect immediately. (count as 2 items).

IV. FUNCTION POINT CALCULATION

The Function Point calculation is a three step process. Step 1 produces the Unadjusted Function Point count, step 2 produces the Value Adjustment Factor (VAF) and step 3 adjusts the Unadjusted Function Point count by the VAF to produce the final Function Point count.

The formula used in step 3 varies depending on the type of count, Application (System Baseline), Development (Project) or Enhancement (Project).

A. Unadjusted Function Point Calculation - Step 1

Unadjusted Function Point values are associated to each identified Function Type based on complexity. There are two types of Function Types, Data and Transactional.

Data Function Types include: Internal Logical Files and External Interface Files. The complexity of each identified Data Function Type is based on the number of Record Types (RET) and Data Element Types (DET).

Transactional Function Types include: External Inputs, External Outputs and External Inquiries. The complexity of each identified Transaction Function Type is based on the number of File Type Referenced (FTR) and Data Element Type (DET).

Once an application's components (specific data and processing requirements) have been categorized into the various function types, each component is assigned an unadjusted Function Point value based its complexity.

The Unadjusted Function Point value for each component is then summarized at the function type level and again at the application level. The resulting total at the application level is the application's Unadjusted Function Point count and is used later in the final calculation.

B. Value Adjustment Factor Calculation

The Value Adjustment Factor is based on 14 General System Characteristics which rate the overall complexity of the application. The 14 General System Characteristics are summarized into the Value Adjustment Factor. When applied, the Value Adjustment Factor adjusts the Unadjusted Function Point count +/- 35 %, producing the final Function Point count.

1) Evaluate the 14 General System Characteristics (GSC) on a scale from 0 to 5 producing a Degree of Influence (DI) for each of the General System Characteristic questions.
2) Sum the 14 DIs to produce the Total Degree of Influence (TDI).
3) Insert the TDI into the following equation to produce the Value Adjustment Factor (VAF):

$$(TDI * 0.01) + 0.65 = VAF$$

Where:

– TDI = the sum of the 14 Degrees of Influence
VAF = the Value Adjustment Factor

The following table can be used to facilitate the calculation of the VAF:

General System Characteristics (GSC)	Degree of Influence (DI) 0 - 5
1. Data Communication	______
2. Distributed Processing	______
3. Performance	______
4. Heavily use configuration	______
5. Transaction rates	______
6. On-line data entry	______
7. Design for end user efficiency	______
8. On-line update	______
9. Complex processing	______
10. Usable in other applications	______
11. Installation ease	______
12. Operational ease	______
13. Multiple sites	______
14. Facilitate change	______
Total Degree of Influence (TDI)	______
Value Adjustment Factor (VAF) (______ X 0.01) + 0.65 = VAF = (TDI * 0.01) + 0.65	______

C. Application Function Point Calculation

To produce the initial Application Function Point count, complete the following formula:

UFP * VAF = AFP

Where:
UFP = the Unadjusted Function Point count
VAF = the Value Adjustment Factor
AFP = the initial Application Function Point count

To produce the revised Application Function Point count, complete the following formula:

((UFPB + ADD + CHGA) - (CHGB + DEL)) * VAFA = AFP

Where:
UFPB = the Application's Unadjusted Function Point prior to the enhancement project.
ADD = the Unadjusted Function Point count of those functions which were added by the enhancement project. CHGA = the Unadjusted Function Point count of those functions which were modified by the enhancement project. This number reflect the functions AFTER the modifications.
CHGB = the Unadjusted Function Point count of those functions which were modified by the enhancement project. This number reflect the functions BEFORE the modifications.
DEL = the Unadjusted Function Point count of those functions which were deleted by the enhancement project.
VAFA = the Value Adjustment Factor of the application after the enhancement project.
AFP = the revised Application Function Point count

D. Development (Project) Function Point calculation

To produce the Development (Project) Function Point count, complete the following formula:

UFP * VAF = DFP

Where:
UFP = the Unadjusted Function Point count
VAF = the Value Adjustment Factor
DFP = the Development (Project) Function Point count

Appendix G - List of Participating Sites

Alberta Wheat Pool
Arbitron
ARCO Oil & Gas Co.
AT&T
CACI
Corp Applications
Databank Systems Ltd.
Development Support Center, Inc.
Eastman Kodak Company - MSD/IS
EG&G Florida Inc.
G.E. Aircraft Engines
G.E. Appliances
G.E. CIT
G.E. Lighting Systems
G.E./AIT/Systems Development
Gouvernement du Quebec
Hughes Aircraft Company
Imperial Life
J.C. Penney
Marine Midland Bank
Montreal Trust
Pacific Bell
Rolls-Royce
Schwans Sales Enterprises
Signet Bank

Form For Rater A

Site No.: ________
Name of Application #1: ______________________________

Name of Application #2: ______________________________

General Instructions. Please prepare Function Point analyses of the two applications selected by your Function Points Evaluation Study Site Coordinator. The analyses should be prepared using, if possible, a requirements definition type of document (e.g. entity-relationship diagrams and process flow diagrams). The analyses must be based on the Function Points Counting Methodology I manual provided to you by the coordinator. Answer all the following questions. Your organization's responses will be kept confidential.

Three other people at your firm are completing this same form. It is very important that you do not discuss the questions with any of these other people. As soon as you have finished answering all the questions, please mail the form in the pre-addressed envelope provided. If you were not given an envelope, please mail the form to:

Function Points Evaluation Study
c/o Prof. Chris F. Kemerer
MIT E53-315
Sloan School of Management
50 Memorial Drive
Cambridge, MA 02139

Analysis of Application 1. Record the Number of Function Counts in the first table below (e.g. in the top left corner of the table, write the number of Low-Complexity External Inputs). Do not multiply by the complexity factors to get Function Points. Next, record the General System Characteristic scores in the second table below.

	Number of Function Counts		
	Low	Average	High
External Input	___	___	___
External Output	___	___	___
Logical Internal File	___	___	___
External Inquiry	___	___	___
External Interface	___	___	___

	General System Characteristic	Score (1-5)
1.	Data communications	_____
2.	Distributed function	_____
3.	Performance	_____
4.	Heavily used configuration	_____
5.	Transaction rate	_____
6.	On-line data entry	_____
7.	Design for end user efficiency	_____
8.	On-line update	_____
9.	Complex processing	_____
10.	Usable in other applications	_____
11.	Installation ease	_____
12.	Operational ease	_____
13.	Multiple sites	_____
14.	Facilitate change	_____

PLEASE TURN OVER --->

Questions About The Analysis of Application 1:

1. How much time did you spend on the analysis of Application 1? ____ hrs
2. What type of basis did you use when making the analysis? (check the one that most closely applies)
    ____ Requirements analysis documents only
    ____ External design documents (hardcopy of example screens, reports, file layouts, etc.) only
    ____ Implemented system
    ____ Other (Please explain): ____

    ____
3. Are you currently doing, or have you ever in the past done any of the development or maintenance of Application 1? (check one of the following):
    ____ Yes, currently
    ____ Yes, previously, but not currently
    ____ No

    3a. If you answered "Yes" on question 3, for how long have you worked on the application? ____ yrs, and ____ months

    3b. If you answered "Yes" on question 3, had you been working mostly full time (i.e. greater than 30 hours per week) or mostly part time on the application? (check one of the following):
    ____ Mostly full-time
    ____ Mostly part-time

    3c. If you answered "No" on question 3, in counting this application, did you have any assistance from personnel who have done some of the development or maintenance of this application? (check one of the following):
    ____ Yes
    ____ No

PLEASE GO ON TO NEXT PAGE

Analysis of Application 2. Record the Number of Function Counts in the first table below (e.g. in the top left corner of the table, write the number of Low-Complexity External Inputs). Do not multiply by the complexity factors to get Function Points. Next, record the General System Characteristic scores in the second table below.

	Number of Function Counts		
	Low	Average	High
External Input	___	___	___
External Output	___	___	___
Logical Internal File	___	___	___
External Inquiry	___	___	___
External Interface	___	___	___

General System Characteristic	Score (1-5)
1. Data communications	_____
2. Distributed function	_____
3. Performance	_____
4. Heavily used configuration	_____
5. Transaction rate	_____
6. On-line data entry	_____
7. Design for end user efficiency	_____
8. On-line update	_____
9. Complex processing	_____
10. Usable in other applications	_____
11. Installation ease	_____
12. Operational ease	_____
13. Multiple sites	_____
14. Facilitate change	_____

Questions About The Analysis of Application 2:

1. How much time did you spend on the analysis of Application 2? ____ hrs
2. What type of basis did you use when making the analysis? (check the one that most closely applies)
   _____ Requirements analysis documents only
   _____ External design documents (hardcopy of example screens, reports, file layouts, etc.) only
   _____ Implemented system
   _____ Other (Please explain): ______________________________________
   ____
3. Are you currently doing, or have you ever in the past done any of the development or maintenance of Application 2? (check one of the following):
   _____ Yes, currently
   _____ Yes, previously, but not currently
   _____ No

   3a. If you answered "Yes" on question 3, for how long have you worked on the application? _____ yrs, and _____ months

   3b. If you answered "Yes" on question 3, had you been working mostly full time (i.e. greater than 30 hours per week) or mostly part time on the application? (check one of the following):
   _____ Mostly full-time
   _____ Mostly part-time

   3c. If you answered "No" on question 3, in counting this application, did you have any assistance from personnel who have done some of the development or maintenance of this application? (check one of the following):
   _____ Yes
   _____ No

PLEASE TURN OVER --->

Background.
1. Over your entire career, how long have you been working in applications development or maintenance? _____ yrs, and _____ months
2. How long have you been working in applications development or maintenance for your current employer? _____ yrs, and _____ months
3. Do you currently use or have you ever used entity-relationship (E-R) modeling in your work? (check one of the following):
   _____ Yes, currently using it.
   _____ Yes, have used it, but currently not using it.
   _____ No, have not used it, but have had some training in it.
   _____ No, have neither used it nor been trained in it.
7. How long have you been using Function Point counting techniques?
   _____ yrs, and _____ months
8. Have you had any formal (classroom) training in Function Point counting? (Check one of the following):
   _____ Yes
   _____ No
   8a. If you answered "Yes" on question 8, how many days of formal training have you received? _____ days
9. Primarily, how did you learn Function Point counting? (check one of the following):
   _____ Internal course
   _____ External course. Taught by:
      _____ IBM
      _____ IFPUG
      _____ Other (name of firm) ________________________________
   _____ Self-Taught
10. Are you personally a member of IFPUG? (check one of the following):
   _____ Yes
   _____ No
11. Have you ever attended an IFPUG conference? (check one of the following):
   _____ Yes
   _____ No

Thank you for your participation.

You have now completed the survey. Please do not discuss the results with your colleagues. Please mail it in the envelope provided. If the envelope is missing, please mail the form to:

Function Points Evaluation Study
c/o Prof. Chris F. Kemerer
MIT E53-315
Sloan School of Management
50 Memorial Drive
Cambridge, MA 02139

Site No.: ________

Introduction

This questionnaire consists of three parts. The first part concerns the applications you selected. The second part concerns your site's Function Point counting conventions. The third part concerns how your site uses Function Points.

We estimate that completing the questionnaire should take one hour. Your responses will be kept confidential.

I. Questions About the Applications

1. Name of Application #1: ______________________________

2. Name of Application #2: ______________________________

3. What type(s) of applications are they? (check whichever of the following most closely applies for each application. Try to avoid the designation "other" unless absolutely necessary):

Ap. #1	Ap. #2	
_____	_____	Batch MIS application
_____	_____	Interactive MIS application
_____	_____	Scientific or mathematical application
_____	_____	Systems software or support application/utility
_____	_____	Communications or telecommunications application
_____	_____	Embedded or real-time application
		Other (Please describe below)
_____	_____	______________________________
_____	_____	______________________________

4. Using a different categorization, what type(s) of applications are they? (check whichever of the following most closely applies for each application. Try to avoid the designation "other" unless absolutely necessary):

Ap. #1	Ap. #2	
_____	_____	Accounting/Finance
_____	_____	Engineering/Design
_____	_____	Human Resources
_____	_____	Manufacturing
_____	_____	Marketing/Sales
_____	_____	Systems Software
_____	_____	End User Tools
		Other (Please describe below)
_____	_____	______________________________
_____	_____	______________________________

5. OPTIONAL. If you have previously made Function Point analyses of these applications using your site's usual counting conventions, how many Function Points were counted?
   Function Points for Application 1: ______
   Function Points for Application 2: ______

6. OPTIONAL. How many work-hours of effort were required to develop these applications? In these work-hours figures, please include the time spent by the IS development team and end-users who were members of the development team. Please exclude the time spent by IS management and non-technical support personnel (e.g. secretaries).
   Application 1: ______ work-hours
   Application 2: ______ work-hours

PLEASE TURN OVER --->

7. OPTIONAL. How many non-comment source lines of code do these applications have in each of the following languages?

Ap. #1	Ap. #2	
______	_____	Ada
______	_____	Assembler
______	_____	C
______	_____	Cobol
______	_____	Fourth Generation Database
______	_____	Fortran
______	_____	Pascal
______	_____	PL/1
		Other (List names below)
______	_____	______________________________
______	_____	______________________________

PLEASE GO TO NEXT PAGE

II. Micro-Cases

In this section, we would like you to answer the questions using your organizations Function Point counting conventions.

1. How does your site count backup files? (check one of the following):
   - _____ Always count them as Logical Internal Files
   - _____ Always count them as External Outputs
   - _____ Count them as Logical Internal Files, but only when backup files are requested by the user and/or auditors
   - _____ Count them as External Outputs, but only when backup files are requested by the user and/or auditors
   - _____ Never count them
   - _____ Other (Please explain): ______________________________

     ____

2. Please refer to the following screen example titled "MultiFunction Address Screen". How many unique External Outputs would your site consider this screen to indicate? Assume that a successful transaction is indicated by displaying a confirmation message on this screen. (check one of the following):
   - _____ One, because the output processing is the same for add, change, and delete functions.
   - _____ Two, because the output processing for the add and change are the same, but the output processing for the delete is different.
   - _____ Three, because add, change, and delete indicate three distinct outputs.
   - _____ Other. (Please explain): ______________________________

     ____

MultiFunction Address Screen

Name: ______________

Address: ______________

City: ______________

State: ___ Zip ______

transaction confirmation message goes here

PF1=Add PF2=Change PF3=Delete

PLEASE TURN OVER --->

3. Please refer to the following screen example titled "Add an Address Screen - I". Assuming two files are referenced, what complexity would your site assign to the External Output associated with this screen? (check one of the following):

   _____ Low. There are five data elements because error messages are not counted.
   _____ Average. There are six data elements because error messages get counted only once as only one message appears on the screen.
   _____ High. There are 25 data elements because each possible error message is counted as an element.
   _____ Other. Please explain: ______________________________

   ____

```
 _______________________________________
|                                       |
|     Add an Address Screen - I         |
|                                       |
|                                       |
|  Name: _______________                |
|  Address: __________________          |
|  City: _______________                |
|  State: __   Zip ______               |
|                                       |
|  error message goes here              |
|_______________________________________|
```

<u>All Possible Error Messages</u> (20 in total)

1. Name too long.
2. Name too short.
3. Not a valid city.
4. Note a valid state.

... etc. ...
... etc. ...

19 Zip code must be numeric.
20. Wrong # digits in zip code.

4. Please refer to the following screen Layout Hierarchy, consisting only of a main menu and five sub-menus, what Function Type(s) would your site use in counting these menus? (check as many as apply):

   _____ Not applicable - menus are not counted
   _____ External Input
   _____ External Output
   _____ Logical Internal File
   _____ External Inquiry
   _____ External Interface

<u>Screen Layout Hierarchy</u>

```
                  |---- Manage Inventory -------
                  |---- Plan Acquisition ---------
Main Menu------   |---- Update Catalogue -------
                  |---- Support Inquiries -------
                  |---- Produce Reports --------
```

5. Referring again to the Screen Layout Hierarchy, how many functions would your site count based on this hierarchy? (check one of the following):

   _____ 0, because menus are not counted
   _____ 1, because menus only get counted once regardless of the number of screens
   _____ 2, because there are two levels
   _____ 6, because there are six menu screens
   _____ Other. Please explain: ______________________________

   ____

PLEASE GO ON TO NEXT PAGE

6. Please refer to the following screen example titled "Add an Address Screen - II". Based on this screen, how many additional functions would your site count due to the help messages? The help message displayed varies depending on the field the cursor is on. (check one of the following): _

   ______ 0, but the complexity rating would reflect the presence of help messages
   ______ 0, but the General Systems Characteristics adjustment would reflect the presence of help messages
   ______ 1, because all help messages are treated as a single function
   ______ 5, because there are 5 help messages
   ______ Other. (Please explain): ______________________________

   ____

```
 _____________________________________
|                                     |
|    Add an Address Screen - II       |
|                                     |
|                                     |
|  Name: ____________                 |
|  Address: ______________            |
|  City: ____________                 |
|  State: __  Zip _____               |
|                                     |
|  help message goes here             |
|_____________________________________|
```

<u>Help Messages</u>
1. Type last name, first name.
2. Address can only be one line.
3. Type name of city.
4. Type 2 character state code.
5. Type 5 or 9 digit zip code.

6a. Referring to the help messages of question 6, how would your site classify the function type for the messages? (check one of the following):

   ______ External Input
   ______ External Outputs
   ______ External Inquiries
   ______ Other. (Please explain): ______________________________

   ____

7. Given the data entry screen of question 6, if there was one help <u>screen</u> per field (rather than a help message per field), how many additional functions would your site count due to the help screens? (check one of the following):

   ______ 0, but the complexity rating would reflect the presence of help screens
   ______ 0, but the General Systems Characteristics adjustment would reflect the presence of help screens
   ______ 1, because all help screens are treated as a single function
   ______ 5, because there are 5 help screens
   ______ Other. (Please explain): ______________________________

   ____

PLEASE TURN OVER --->

7a. Referring to the help screens of question 7, how would your site classify the function type for the screens? (check one of the following):

_____ Internal Logical Files
_____ External Interface Files
_____ External Input
_____ External Outputs
_____ External Inquiries
_____ Other. (Please explain): ____

____

8. Assume a report with detail lines, subtotals, and a grand total, where all lines have the same format. At your site, would you count this as:

_____ One External Output, with the subtotals and grand totals adding to the number of data elements.
_____ Two External Outputs: one including only the detail lines, and another including only the subtotals and grand totals.
_____ Three External Outputs: one including only the detail lines, another including only the subtotals, and another including only the grand totals.
_____ Other. (Please explain): ____

____

9. What function type does your site use for hard coded tables (i.e. tables which only a programmer, and not an end-user can change)? (check one of the following):

_____ Logical Internal Files, because they are files
_____ External Interfaces
_____ None, because they are not user-changeable
_____ Other. (Please explain): ____

____

10. Please refer to the following report layout titled "Customer Orders". Assume that this report can be produced with either of two selection criteria: by selecting dates or by selecting customer numbers. The data is ordered (sorted) by customer number regardless of the selection criteria used. How many External Outputs would your site count this report as? (check one of the following):

_____ One, because the report format is the same for both selection criteria
_____ Two, because the data is different depending on the selection criteria
_____ Other. (Please explain): ____

____

Customer Orders

Cust #	Part #	Order Date	Quantity
1111	1111	1/1/88	11
2222	2222	2/2/89	22
3333	3333	3/3/89	33

PLEASE GO TO NEXT PAGE

11. Referring again to the report layout titled "Customer Orders". Assume that this report can be ordered (sorted) with either of two criteria: by date or by customer numbers. How many external outputs would your site count this report as? (check one of the following):

_____ One, because the report format is the same for both ordering criteria
_____ Two, because the data is different depending on the ordering criteria
_____ Other. (Please explain): ______________________________

____

12. For External Inquiries, which of the following sets of function point weights does your site use for low, average, and high complexity? (check one of the following):

_____ Three for Simple, Four for Average, Six for Complex
_____ Four for Simple, Five for Average, Six or Seven for Complex
_____ Other. Please describe: _____ Simple, _____ Average, _____ Complex

13. If Application A reads one of Application B's Logical Internal Files and converts the data into transactions to update one of its own Logical Internal Files, how would your site classify the Logical Internal File in Application B? (check one of the following)?

_____ As a Logical Internal File and an External Interface File
_____ As a Logical Internal File and an External Output File
_____ Only as a logical Internal File
_____ Other. (Please explain): ______________________________

____

14 If Application A creates a file of transaction data from Application B's Logical Internal File, how would your site classify Application A's transaction file? (check one of the following):

_____ As an External Input
_____ As am External Interface File
_____ As a Logical Internal File
_____ As nothing (i.e. it would not be counted), because it is a temporary file.
_____ Other (Please explain): ______________________________

____

PLEASE TURN OVER --->

III. Site Questions.

1. What industry is your site in (check the one that most closely applies):
   - ______ Conglomerate
   - ______ Agriculture, Forestry & Fishing
   - ______ Mining
   - ______ Construction
   - ______ Manufacturing
   - ______ Transportation, Communication, Electric, Gas & Sanitary
   - ______ Wholesale & Retail Trade
   - ______ Finance, Insurance & Real Estate
   - ______ Services
   - ______ Government
2. How long has your site been using Function Point analysis? _____ yrs, and _____ months
3. Which of the following Function Point methodologies does your site's counting practices most closely resemble? (check one of the following):
   - ______ IBM/Albrecht 1979
   - ______ IBM/GUIDE 1984
   - ______ IFPUG Counting Practices Manual 2.0 or 2.1
   - ______ Charles Symons Mark II
   - ______ Software Productivity Research Function Points
   - ______ Software Productivity Research Feature Points
   - ______ Brian Dreger, Function Point Analysis (1989)
   - ______ Donald Reifer Real-Time extension
   - ______ Other (Please describe): ______________________________

     ____
4. Does your site use any automated tool to assist in counting Function Points?
   - ______ Yes, automated tool used
   - ______ No, manual only

   4a. If you answered "Yes" to question 4, what system(s) do you use? (check all of the following that are used):
   - ______ DMR Expert System
   - ______ Qualitative Software Management PADS
   - ______ Rubin ESTIMACS
   - ______ SPR Checkpoint
   - ______ Texas Instruments IEF
   - ______ Other (Please describe): ______________________________

     ____
5. How have you documented counting rules at your site? (check all of the following that are used):
   - _____ Training is required. Minimum training hours: ______
   - _____ Internally developed counting manual used by all counters. Number of pages: _____
   - _____ Externally developed counting manual used by all counters. Please name: ________

PLEASE GO TO NEXT PAGE

6. What do you use Function Points for? (check as many boxes as apply):

	New Development	Maintenance
Manpower estimating	___	___
Change impact estimating	___	___
Progress measurement	___	___
Productivity measurement	___	___
Management reporting	___	___
Other (Please list below)	___	___

Other uses: ____

____

7. If you use Function Points for estimating development effort, do you add contingency to the resulting manpower estimates?
   _____ Yes
   _____ No

   7a. If you answered "Yes" in question 7, how much contingency would you typically add to an analysis made from a requirements definition document? (e.g. if the Function Point analysis predicted a 20 work-month project, and your final estimate is 25 work-months, then your contingency is (25-20)/20 = 25%.) ____%

8. Please provide any suggestions or comments you may have for the IFPUG Counting Practices Committee in the space below.

____

____

____

____

____

____

____

____

____

Thank you for completing this survey. Please mail it in the envelope provided. If the envelope is missing, please mail it to:

Function Points Evaluation Study
c/o Prof. Chris F. Kemerer
MIT E53-315
Sloan School of Management
50 Memorial Drive
Cambridge, MA 02139

Thank you for your participation.

Background.

1. Over your entire career, how long have you been working in applications development or maintenance? _____ yrs, and _____ months
2. How long have you been working in applications development or maintenance for your current employer? _____ yrs, and _____ months
3. Do you currently use or have you ever used entity-relationship (E-R) modeling in your work? (check one of the following):
   - 19 Yes, currently using it.
   - 14 Yes, have used it, but currently not using it.
   - 21 No, have not used it, but have had some training in it.
   - 27 No, have neither used it nor been trained in it.
7. How long have you been using Function Point counting techniques? _____ yrs, and _____ months
8. Have you had any formal (classroom) training in Function Point counting? (Check one of the following):
   - 59 Yes
   - 19 No

   8a. If you answered "Yes" on question 8, how many days of formal training have you received? _____ days
9. Primarily, how did you learn Function Point counting? (check one of the following):
   - 38 Internal course
   - 27 External course. Taught by:
     - 4 IBM
     - 9 IFPUG
     - 17 Other (name of firm) ____________________
   - 25 Self-Taught
10. Are you personally a member of IFPUG? (check one of the following):
    - 21 Yes
    - 60 No
11. Have you ever attended an IFPUG conference? (check one of the following):
    - 29 Yes
    - 52 No

Thank you for your participation.

You have now completed the survey. Please do not discuss the results with your colleagues. Please mail it in the envelope provided. If the envelope is missing, please mail the form to:

Function Points Evaluation Study
c/o Prof. Chris F. Kemerer
MIT E53-315
Sloan School of Management
50 Memorial Drive
Cambridge, MA 02139

Questionnaire for Coordinator

Site No.: ________

Introduction

This questionnaire consists of three parts. The first part concerns the applications you selected. The second part concerns your site's Function Point counting conventions. The third part concerns how your site uses Function Points.

We estimate that completing the questionnaire should take one hour. Your responses will be kept confidential.

I. Questions About the Applications

1. Name of Application #1: ____________________

2. Name of Application #2: ____________________

3. What type(s) of applications are they? (check whichever of the following most closely applies for each application. Try to avoid the designation "other" unless absolutely necessary):

Ap. #1	Ap. #2	
7	____	Batch MIS application
33	____	Interactive MIS application
0	____	Scientific or mathematical application
4	____	Systems software or support application/utility
0	____	Communications or telecommunications application
0	____	Embedded or real-time application
		Other (Please describe below)
0	____	____________________
____	____	____________________

4. Using a different categorization, what type(s) of applications are they? (check whichever of the following most closely applies for each application. Try to avoid the designation "other" unless absolutely necessary):

Ap. #1	Ap. #2	
19	____	Accounting/Finance
2	____	Engineering/Design
0	____	Human Resources
9	____	Manufacturing
4	____	Marketing/Sales
1	____	Systems Software
2	____	End User Tools
		Other (Please describe below)
7	____	____________________
____	____	____________________

5. OPTIONAL. If you have previously made Function Point analyses of these applications using your site's usual counting conventions, how many Function Points were counted?
   Function Points for Application 1: ______
   Function Points for Application 2: ______

6. OPTIONAL. How many work-hours of effort were required to develop these applications? In these work-hours figures, please include the time spent by the IS development team and end-users who were members of the development team. Please exclude the time spent by IS management and non-technical support personnel (e.g. secretaries).
   Application 1: ______ work-hours
   Application 2: ______ work-hours

PLEASE TURN OVER --->

In this section, we would like you to answer the questions using your organizations Function Point counting conventions.

1. How does your site count backup files? (check one of the following):
   - 1 Always count them as Logical Internal Files
   - 2 Always count them as External Outputs
   - 18 Count them as Logical Internal Files, but only when backup files are requested by the user and/or auditors
   - 7 Count them as External Outputs, but only when backup files are requested by the user and/or auditors
   - 12 Never count them
   - 5 Other (Please explain): ____________

2. Please refer to the following screen example titled "MultiFunction Address Screen". How many unique External Outputs would your site consider this screen to indicate? Assume that a successful transaction is indicated by displaying a confirmation message on this screen. (check one of the following):
   - 13 One, because the output processing is the same for add, change, and delete functions.
   - 3 Two, because the output processing for the add and change are the same, but the output processing for the delete is different.
   - 19 Three, because add, change, and delete indicate three distinct outputs.
   - 10 Other. (Please explain): ____________

MultiFunction Address Screen

Name: ____________
Address: ____________
City: ____________
State: ___ Zip ______

transaction confirmation message goes here

PF1=Add PF2=Change PF3=Delete

PLEASE TURN OVER --->

3. Please refer to the following screen example titled "Add an Address Screen - I". Assuming two files are referenced, what complexity would your site assign to the External Output associated with this screen? (check one of the following):

   6 Low. There are five data elements because error messages are not counted.

   14 Average. There are six data elements because error messages get counted only once as only one message appears on the screen.

   9 High. There are 25 data elements because each possible error message is counted as an element.

   15 Other. Please explain: ______________________________

```
 ___________________________________
|                                   |
|   Add an Address Screen - I       |
|                                   |
|                                   |
| Name: ______________              |
| Address: _________________        |
| City: ______________              |
| State: ___   Zip _____            |
|                                   |
| error message goes here           |
|___________________________________|
```

All Possible Error Messages (20 in total)

1. Name too long.
2. Name too short.
3. Not a valid city.
4. Note a valid state.

... etc. ...

... etc. ...

19 Zip code must be numeric.

20. Wrong # digits in zip code.

4. Please refer to the following screen Layout Hierarchy, consisting only of a main menu and five sub-menus, what Function Type(s) would your site use in counting these menus? (check as many as apply):

   16 Not applicable - menus are not counted

   3 External Input

   1 External Output

   1 Logical Internal File

   22 External Inquiry

   0 External Interface

Screen Layout Hierarchy

```
              |---- Manage Inventory -------
              |---- Plan Acquisition ---------
Main Menu------ |---- Update Catalogue -------
              |---- Support Inquiries -------
              |---- Produce Reports --------
```

5. Referring again to the Screen Layout Hierarchy, how many functions would your site count based on this hierarchy? (check one of the following):

   16 0, because menus are not counted

   7 1, because menus only get counted once regardless of the number of screens

   2 2, because there are two levels

   17 6, because there are six menu screens

   1 Other. Please explain: ______________________________

.PLEASE GO ON TO NEXT PAGE

6. Please refer to the following screen example titled "Add an Address Screen - II". Based on this screen, how many additional functions would your site count due to the help messages? The help message displayed varies depending on the field the cursor is on. (check one of the following):

3	0, but the complexity rating would reflect the presence of help messages
3	0, but the General Systems Characteristics adjustment would reflect the presence of help messages
27	1, because all help messages are treated as a single function
5	5, because there are 5 help messages
6	Other. (Please explain): ______________________

Add an Address Screen - II

Name: ______________
Address: ________________
City: ______________
State: ___ Zip ______

help message goes here

Help Messages
1. Type last name, first name.
2. Address can only be one line.
3. Type name of city.
4. Type 2 character state code.
5. Type 5 or 9 digit zip code.

6a. Referring to the help messages of question 6, how would your site classify the function type for the messages? (check one of the following):

0	External Input
13	External Outputs
28	External Inquiries
3	Other. (Please explain): ______________________

7. Given the data entry screen of question 6, if there was one help screen per field (rather than a help message per field), how many additional functions would your site count due to the help screens? (check one of the following):

2	0, but the complexity rating would reflect the presence of help screens
3	0, but the General Systems Characteristics adjustment would reflect the presence of help screens
22	1, because all help screens are treated as a single function
14	5, because there are 5 help screens
2	Other. (Please explain): ______________________

PLEASE TURN OVER --->

7a. Referring to the help screens of question 7, how would your site classify the function type for the screens? (check one of the following):

- 1 Internal Logical Files
- 0 External Interface Files
- 1 External Input
- 10 External Outputs
- 29 External Inquiries
- 2 Other. (Please explain): ______

8. Assume a report with detail lines, subtotals, and a grand total, where all lines have the same format. At your site, would you count this as:

- 39 One External Output, with the subtotals and grand totals adding to the number of data elements.
- 2 Two External Outputs: one including only the detail lines, and another including only the subtotals and grand totals.
- 2 Three External Outputs: one including only the detail lines, another including only the subtotals, and another including only the grand totals.
- 1 Other. (Please explain): ______

9. What function type does your site use for hard coded tables (i.e. tables which only a programmer, and not an end-user can change)? (check one of the following):

- 13 Logical Internal Files, because they are files
- 3 External Interfaces
- 23 None, because they are not user-changeable
- 5 Other. (Please explain): ______

10. Please refer to the following report layout titled "Customer Orders". Assume that this report can be produced with either of two <u>selection criteria</u>: by selecting dates or by selecting customer numbers. The data is ordered (sorted) by customer number regardless of the selection criteria used. How many External Outputs would your site count this report as? (check one of the following):

- 26 One, because the report format is the same for both selection criteria
- 16 Two, because the data is different depending on the selection criteria
- 2 Other. (Please explain): ______

Customer Orders

Cust #	Part #	Order Date	Quantity
1111	1111	1/1/88	11
2222	2222	2/2/89	22
3333	3333	3/3/89	33

PLEASE GO TO NEXT PAGE

11. Referring again to the report layout titled "Customer Orders". Assume that this report can be ordered (sorted) with either of two criteria: by date or by customer numbers. How many external outputs would your site count this report as? (check one of the following):

- 20 One, because the report format is the same for both ordering criteria
- 23 Two, because the data is different depending on the ordering criteria
- 2 Other. (Please explain): ____

12. For External Inquiries, which of the following sets of function point weights does your site use for low, average, and high complexity? (check one of the following):

- 40 Three for Simple, Four for Average, Six for Complex
- 1 Four for Simple, Five for Average, Six or Seven for Complex
- 2 Other. Please describe: ____ Simple, ____ Average, ____ Complex

13. If Application A reads one of Application B's Logical Internal Files and converts the data into transactions to update one of its own Logical Internal Files, how would your site classify the Logical Internal File in Application B? (check one of the following)?

- 17 As a Logical Internal File and an External Interface File
- 4 As a Logical Internal File and an External Output File
- 15 Only as a logical Internal File
- 9 Other. (Please explain): ____

14 If Application A creates a file of transaction data from Application B's Logical Internal File, how would your site classify Application A's transaction file? (check one of the following):

- 16 As an External Input
- 6 As am External Interface File
- 5 As a Logical Internal File
- 13 As nothing (i.e. it would not be counted), because it is a temporary file.
- 5 Other (Please explain): ____

PLEASE TURN OVER --->

III. Site Questions.

1. What industry is your site in (check the one that most closely applies):
   - 4 Conglomerate
   - 1 Agriculture, Forestry & Fishing
   - 1 Mining
   - 0 Construction
   - 10 Manufacturing
   - 7 Transportation, Communication, Electric, Gas & Sanitary
   - 3 Wholesale & Retail Trade
   - 10 Finance, Insurance & Real Estate
   - 6 Services
   - 2 Government
2. How long has your site been using Function Point analysis? _____ yrs, and _____ months
3. Which of the following Function Point methodologies does your site's counting practices most closely resemble? (check one of the following):
   - 12 IBM/Albrecht 1979
   - 11 IBM/GUIDE 1984
   - 18 IFPUG Counting Practices Manual 2.0 or 2.1
   - 0 Charles Symons Mark II
   - 0 Software Productivity Research Function Points
   - 0 Software Productivity Research Feature Points
   - 0 Brian Dreger, Function Point Analysis (1989)
   - 0 Donald Reifer Real-Time extension
   - 3 Other (Please describe): ____________________
4. Does your site use any automated tool to assist in counting Function Points?
   - 22 Yes, automated tool used
   - 22 No, manual only

   4a. If you answered "Yes" to question 4, what system(s) do you use? (check all of the following that are used):
   - 1 DMR Expert System
   - 0 Qualitative Software Management PADS
   - 2 Rubin ESTIMACS
   - 2 SPR Checkpoint
   - 1 Texas Instruments IEF
   - 15 Other (Please describe): ____________________
5. How have you documented counting rules at your site? (check all of the following that are used):
   - 26 Training is required. Minimum training hours: ______
   - 26 Internally developed counting manual used by all counters. Number of pages: _____
   - 2 Externally developed counting manual used by all counters. Please name: ________

PLEASE GO TO NEXT PAGE

6. What do you use Function Points for? (check as many boxes as apply):

	New Development	Maintenance
Manpower estimating	23	18
Change impact estimating	12	10
Progress measurement	15	7
Productivity measurement	39	27
Management reporting	32	24
Other (Please list below)	7	7

Other uses: ______________________________

____

7. If you use Function Points for estimating development effort, do you add contingency to the resulting manpower estimates?
   10 Yes
   16 No

   7a. If you answered "Yes" in question 7, how much contingency would you typically add to an analysis made from a requirements definition document? (e.g. if the Function Point analysis predicted a 20 work-month project, and your final estimate is 25 work-months, then your contingency is (25-20)/20 = 25%.) ____%

8. Please provide any suggestions or comments you may have for the IFPUG Counting Practices Committee in the space below.

____

____

____

____

____

____

____

____

____

Thank you for completing this survey. Please mail it in the envelope provided. If the envelope is missing, please mail it to:

Function Points Evaluation Study
c/o Prof. Chris F. Kemerer
MIT E53-315
Sloan School of Management
50 Memorial Drive
Cambridge, MA 02139

Thank you for your participation.

_ Appendix N - Statistical Analysis Programs

This program reads in the raw data and creates numerous SAS datasets which are analyzed by other programs.

```
CMS FILEDEF FPANAL1  DISK FP_ANAL1 DAT A;
CMS FILEDEF FPANAL2  DISK FP_ANAL2 DAT A;
CMS FILEDEF RATERBK  DISK RATER_BK DAT A;
CMS FILEDEF APPLICAT DISK APPLICAT DAT A;
CMS FILEDEF LANGUAGE DISK LANGUAGE DAT A;
CMS FILEDEF MICROCA  DISK MICRO_CA DAT A;
CMS FILEDEF SITEQUS  DISK SITEQUS DAT A;
*;
*  ============================================================  */;
*;
DATA FP.FPANAL1                /*  READ DATA FROM FP_ANAL1 DAT  */;
INFILE FPANAL1                 /*  KEY IS SITE/RATER/APPLIC#  */;
INPUT
      @1  SITE__NO  2.
      @4  RATER_ID $1.
      @6  APPLC_NO  1.
      @8  INPUTS_L  3.
      @12 INPUTS_M  3.
      @16 INPUTS_H  3.
      @20 OUTPUT_L  3.
      @24 OUTPUT_M  3.
      @28 OUTPUT_H  3.
      @32 FILES_LO  3.
      @36 FILES_ME  3.
      @40 FILES_HI  3.
      @44 INQUIR_L  3.
      @48 INQUIR_M  3.
      @52 INQUIR_H  3.
      @56 INTERF_L  3.
      @60 INTERF_M  3.
      @64 INTERF_H  3.;
                               /*  CALCULATE FP COUNTS PER TYPE */;
TOT_INPT = 3 * INPUTS_L +  4 * INPUTS_M +  6 * INPUTS_H;
TOT_OUTP = 4 * OUTPUT_L +  5 * OUTPUT_M +  7 * OUTPUT_H;
TOT_FILE = 7 * FILES_LO + 10 * FILES_ME + 15 * FILES_HI;
TOT_INQU = 3 * INQUIR_L +  4 * INQUIR_M +  6 * INQUIR_H;
TOT_INTR = 5 * INTERF_L +  7 * INTERF_M + 10 * INTERF_H;
GDTOT_FP = TOT_INPT + TOT_OUTP + TOT_FILE + TOT_INQU + TOT_INTR;
*;
LENGTH TEMPVAR1 $ 2            /*  ADD MULTI-COLUMN KEYS  */;
TEMPVAR1 = SITE__NO;
LENGTH KEY_SR $ 3;
KEY_SR = TEMPVAR1 || RATER_ID;
LENGTH TEMPVAR2 $ 1;
TEMPVAR2 = APPLC_NO;
LENGTH KEY_SRA $ 4;
KEY_SRA = KEY_SR || TEMPVAR2;
LENGTH KEY_SA $ 3;
KEY_SA = TEMPVAR1 || TEMPVAR2;
DROP TEMPVAR1 TEMPVAR2;
```

```
*;
* ==================================================================  ;
*;
DATA FPANAL1A                  /*  TEMPORARY DATASETS  */;
SET FP.FPANAL1                 /*  USED TO CREATE FPANAL1T  */;
IF RATER_ID = 'A';
A_INPU_L = INPUTS_L;
A_INPU_M = INPUTS_M;
A_INPU_H = INPUTS_H;
A_OUTP_L = OUTPUT_L;
A_OUTP_M = OUTPUT_M;
A_OUTP_H = OUTPUT_H;
A_FILE_L = FILES_LO;
A_FILE_M = FILES_ME;
A_FILE_H = FILES_HI;
A_INQU_L = INQUIR_L;
A_INQU_M = INQUIR_M;
A_INQU_H = INQUIR_H;
A_INTR_L = INTERF_L;
A_INTR_M = INTERF_M;
A_INTR_H = INTERF_H;
A_TO_INP = TOT_INPT;
A_TO_OUT = TOT_OUTP;
A_TO_FIL = TOT_FILE;
A_TO_INQ = TOT_INQU;
A_TO_INT = TOT_INTR;
A_FC = GDTOT_FP;
*;
DATA FPANAL1B;
SET FP.FPANAL1;
IF RATER_ID = 'B';
B_INPU_L = INPUTS_L;
B_INPU_M = INPUTS_M;
B_INPU_H = INPUTS_H;
B_OUTP_L = OUTPUT_L;
B_OUTP_M = OUTPUT_M;
B_OUTP_H = OUTPUT_H;
B_FILE_L = FILES_LO;
B_FILE_M = FILES_ME;
B_FILE_H = FILES_HI;
B_INQU_L = INQUIR_L;
B_INQU_M = INQUIR_M;
B_INQU_H = INQUIR_H;
B_INTR_L = INTERF_L;
B_INTR_M = INTERF_M;
B_INTR_H = INTERF_H;
B_TO_INP = TOT_INPT;
B_TO_OUT = TOT_OUTP;
B_TO_FIL = TOT_FILE;
B_TO_INQ = TOT_INQU;
B_TO_INT = TOT_INTR;
B_FC = GDTOT_FP;
*;
DATA FPANAL1C;
SET FP.FPANAL1;
IF RATER_ID = 'C';
C_INPU_L = INPUTS_L;
```

```
C_INPU_M = INPUTS_M;
C_INPU_H = INPUTS_H;
C_OUTP_L = OUTPUT_L;
C_OUTP_M = OUTPUT_M;
C_OUTP_H = OUTPUT_H;
C_FILE_L = FILES_LO;
C_FILE_M = FILES_ME;
C_FILE_H = FILES_HI;
C_INQU_L = INQUIR_L;
C_INQU_M = INQUIR_M;
C_INQU_H = INQUIR_H;
C_INTR_L = INTERF_L;
C_INTR_M = INTERF_M;
C_INTR_H = INTERF_H;
C_TO_INP = TOT_INPT;
C_TO_OUT = TOT_OUTP;
C_TO_FIL = TOT_FILE;
C_TO_INQ = TOT_INQU;
C_TO_INT = TOT_INTR;
C_FC = GDTOT_FP;
*;
DATA FPANAL1D;
SET FP.FPANAL1;
IF RATER_ID = 'D';
D_INPU_L = INPUTS_L;
D_INPU_M = INPUTS_M;
D_INPU_H = INPUTS_H;
D_OUTP_L = OUTPUT_L;
D_OUTP_M = OUTPUT_M;
D_OUTP_H = OUTPUT_H;
D_FILE_L = FILES_LO;
D_FILE_M = FILES_ME;
D_FILE_H = FILES_HI;
D_INQU_L = INQUIR_L;
D_INQU_M = INQUIR_M;
D_INQU_H = INQUIR_H;
D_INTR_L = INTERF_L;
D_INTR_M = INTERF_M;
D_INTR_H = INTERF_H;
D_TO_INP = TOT_INPT;
D_TO_OUT = TOT_OUTP;
D_TO_FIL = TOT_FILE;
D_TO_INQ = TOT_INQU;
D_TO_INT = TOT_INTR;
D_FC = GDTOT_FP;
*;
DATA FP.FPANAL1T                /*  KEY IS SITE/APPLICATION  */;
                                /*  DATA BY RATER IS IN CLUSTERS  */;
                                /*  OF COLUMNS  */;
MERGE FPANAL1A FPANAL1B FPANAL1C FPANAL1D;
BY KEY_SA;
DROP RATER_ID KEY_SR KEY_SRA SITE__NO
      INPUTS_L INPUTS_M INPUTS_H OUTPUT_L OUTPUT_M OUTPUT_H
      FILES_LO FILES_ME FILES_HI INQUIR_L INQUIR_M INQUIR_H
      INTERF_L INTERF_M INTERF_H
      TOT_INPT TOT_OUTP TOT_FILE TOT_INQU TOT_INTR GDTOT_FP;
AVAB_INP = (A_TO_INP + B_TO_INP) / 2;
```

```
AVCD_INP = (C_TO_INP + D_TO_INP) / 2;
AVAB_OUT = (A_TO_OUT + B_TO_OUT) / 2;
AVCD_OUT = (C_TO_OUT + D_TO_OUT) / 2;
AVAB_FIL = (A_TO_FIL + B_TO_FIL) / 2;
AVCD_FIL = (C_TO_FIL + D_TO_FIL) / 2;
AVAB_INQ = (A_TO_INQ + B_TO_INQ) / 2;
AVCD_INQ = (C_TO_INQ + D_TO_INQ) / 2;
AVAB_INT = (A_TO_INT + B_TO_INT) / 2;
AVCD_INT = (C_TO_INT + D_TO_INT) / 2;
AVGAB_FC = (A_FC + B_FC) / 2;
AVGCD_FC = (C_FC + D_FC) / 2;
IF AVAB_INP NE . AND AVAB_INP NE 0 THEN DO;
     MRE_AINP = ABS((AVAB_INP - A_TO_INP) / AVAB_INP);
     MRE_BINP = ABS((AVAB_INP - B_TO_INP) / AVAB_INP);
     END;
IF AVCD_INP NE . AND AVCD_INP NE 0 THEN DO;
     MRE_CINP = ABS((AVCD_INP - C_TO_INP) / AVCD_INP);
     MRE_DINP = ABS((AVCD_INP - D_TO_INP) / AVCD_INP);
     END;
IF AVAB_OUT NE . AND AVAB_OUT NE 0 THEN DO;
     MRE_AOUT = ABS((AVAB_OUT - A_TO_OUT) / AVAB_OUT);
     MRE_BOUT = ABS((AVAB_OUT - B_TO_OUT) / AVAB_OUT);
     END;
IF AVCD_OUT NE . AND AVCD_OUT NE 0 THEN DO;
     MRE_COUT = ABS((AVCD_OUT - C_TO_OUT) / AVCD_OUT);
     MRE_DOUT = ABS((AVCD_OUT - D_TO_OUT) / AVCD_OUT);
     END;
IF AVAB_FIL NE . AND AVAB_FIL NE 0 THEN DO;
     MRE_AFIL = ABS((AVAB_FIL - A_TO_FIL) / AVAB_FIL);
     MRE_BFIL = ABS((AVAB_FIL - B_TO_FIL) / AVAB_FIL);
     END;
IF AVCD_FIL NE . AND AVCD_FIL NE 0 THEN DO;
     MRE_CFIL = ABS((AVCD_FIL - C_TO_FIL) / AVCD_FIL);
     MRE_DFIL = ABS((AVCD_FIL - D_TO_FIL) / AVCD_FIL);
     END;
IF AVAB_INQ NE . AND AVAB_INQ NE 0 THEN DO;
     MRE_AINQ = ABS((AVAB_INQ - A_TO_INQ) / AVAB_INQ);
     MRE_BINQ = ABS((AVAB_INQ - B_TO_INQ) / AVAB_INQ);
     END;
IF AVCD_INQ NE . AND AVCD_INQ NE 0 THEN DO;
     MRE_CINQ = ABS((AVCD_INQ - C_TO_INQ) / AVCD_INQ);
     MRE_DINQ = ABS((AVCD_INQ - D_TO_INQ) / AVCD_INQ);
     END;
IF AVAB_INT NE . AND AVAB_INT NE 0 THEN DO;
     MRE_AINT = ABS((AVAB_INT - A_TO_INT) / AVAB_INT);
     MRE_BINT = ABS((AVAB_INT - B_TO_INT) / AVAB_INT);
     END;
IF AVCD_INT NE . AND AVCD_INT NE 0 THEN DO;
     MRE_CINT = ABS((AVCD_INT - C_TO_INT) / AVCD_INT);
     MRE_DINT = ABS((AVCD_INT - D_TO_INT) / AVCD_INT);
     END;
IF AVGAB_FC NE . AND AVGAB_FC NE 0 THEN DO;
     MRE_A_FC = ABS((AVGAB_FC - A_FC) / AVGAB_FC);
     MRE_B_FC = ABS((AVGAB_FC - B_FC) / AVGAB_FC);
     END;
IF AVGCD_FC NE . AND AVGCD_FC NE 0 THEN DO;
     MRE_C_FC = ABS((AVGCD_FC - C_FC) / AVGCD_FC);
```

```
     MRE_D_FC = ABS((AVGCD_FC - D_FC) / AVGCD_FC);
     END;
                                    /* IF ONLY 1 A,B OR C,D USE THE 1 */;
IF A_TO_INP = . THEN AVAB_INP = B_TO_INP;
IF B_TO_INP = . THEN AVAB_INP = A_TO_INP;
IF C_TO_INP = . THEN AVCD_INP = D_TO_INP;
IF D_TO_INP = . THEN AVCD_INP = C_TO_INP;
IF A_TO_OUT = . THEN AVAB_OUT = B_TO_OUT;
IF B_TO_OUT = . THEN AVAB_OUT = A_TO_OUT;
IF C_TO_OUT = . THEN AVCD_OUT = D_TO_OUT;
IF D_TO_OUT = . THEN AVCD_OUT = C_TO_OUT;
IF A_TO_FIL = . THEN AVAB_FIL = B_TO_FIL;
IF B_TO_FIL = . THEN AVAB_FIL = A_TO_FIL;
IF C_TO_FIL = . THEN AVCD_FIL = D_TO_FIL;
IF D_TO_FIL = . THEN AVCD_FIL = C_TO_FIL;
IF A_TO_INQ = . THEN AVAB_INQ = B_TO_INQ;
IF B_TO_INQ = . THEN AVAB_INQ = A_TO_INQ;
IF C_TO_INQ = . THEN AVCD_INQ = D_TO_INQ;
IF D_TO_INQ = . THEN AVCD_INQ = C_TO_INQ;
IF A_TO_INT = . THEN AVAB_INQ = B_TO_INQ;
IF B_TO_INT = . THEN AVAB_INQ = A_TO_INQ;
IF C_TO_INT = . THEN AVCD_INQ = D_TO_INQ;
IF D_TO_INT = . THEN AVCD_INQ = C_TO_INQ;
IF A_FC = . THEN AVGAB_FC = B_FC;
IF B_FC = . THEN AVGAB_FC = A_FC;
IF C_FC = . THEN AVGCD_FC = D_FC;
IF D_FC = . THEN AVGCD_FC = C_FC;
*;
*  ============================================================  */;
*;
DATA FP.FPANAL2                     /*  INPUT DATA FROM FP_ANAL2 DAT  */;
INFILE FPANAL2                      /*  KEY IS SITE/RATER/APPLIC#  */;
INPUT
      @1  SITE__NO  2.
      @4  RATER_ID $1.
      @6  APPLC_NO  1.
      @8  DATA_COM  1.
      @10 DIST_FUN  1.
      @12 PERFORMA  1.
      @14 HEAVY_US  1.
      @16 TRANS_RT  1.
      @18 ONLINE_E  1.
      @20 END_USER  1.
      @22 ONLINE_U  1.
      @24 COMPLEXP  1.
      @26 USE_OTHR  1.
      @28 INSTALLE  1.
      @30 OPER_EAS  1.
      @32 MULTI_ST  1.
      @34 FACIL_CG  1.
      @36 TIME_SPT  4.1
      @41 BASIS_TP  1.
      @43 OTHR_BAS $1.
      @45 WORK_APP  1.
      @47 WORK_LGT  4.1
      @52 WORK_TIM  1.
      @54 WORK_ASS  1.;
```

```
*;
LENGTH SUM_CHAR 3.              /*  CALCULATE FP ADJUSTMENT  */;
SUM_CHAR = DATA_COM + DIST_FUN + PERFORMA + HEAVY_US + TRANS_RT +
      ONLINE_E + END_USER + COMPLEXP + USE_OTHR + INSTALLE +
      OPER_EAS + MULTI_ST + FACIL_CG;
GSC_ADJU = SUM_CHAR * 0.01 + 0.65;
*;
LENGTH TEMPVAR1 $ 2             /*  ADD MULTI-COLUMN KEYS  */;
TEMPVAR1 = SITE__NO;
LENGTH KEY_SR $ 3;
KEY_SR = TEMPVAR1 || RATER_ID;
LENGTH TEMPVAR2 $ 1;
TEMPVAR2 = APPLC_NO;
LENGTH KEY_SRA $ 4;
KEY_SRA = KEY_SR || TEMPVAR2;
LENGTH KEY_SA $3;
KEY_SA = TEMPVAR1 || TEMPVAR2;
DROP TEMPVAR1 TEMPVAR2;
*;
* ============================================================  */;
*;
DATA FPANAL2A                   /*  TEMPORARY DATASETS */;
SET FP.FPANAL2                  /*  USED TO CREATE FPANAL2T  */;
IF RATER_ID = 'A';
A_DATA_C = DATA_COM;
A_DIST_F = DIST_FUN;
A_PERFOR = PERFORMA;
A_HEAVYU = HEAVY_US;
A_TRAN_R = TRANS_RT;
A_ONLINE = ONLINE_E;
A_END_US = END_USER;
A_ONLINU = ONLINE_U;
A_COMP_P = COMPLEXP;
A_USE_OT = USE_OTHR;
A_INSTAL = INSTALLE;
A_OPER_E = OPER_EAS;
A_MULTIS = MULTI_ST;
A_FACILC = FACIL_CG;
A_SUMCHA = SUM_CHAR;
A_GSC_AD = GSC_ADJU;
A_BASIST = BASIS_TP;
A_WORK_A = WORK_APP;
*;
DATA FPANAL2B;
SET FP.FPANAL2;
IF RATER_ID = 'B';
B_DATA_C = DATA_COM;
B_DIST_F = DIST_FUN;
B_PERFOR = PERFORMA;
B_HEAVYU = HEAVY_US;
B_TRAN_R = TRANS_RT;
B_ONLINE = ONLINE_E;
B_END_US = END_USER;
B_ONLINU = ONLINE_U;
B_COMP_P = COMPLEXP;
B_USE_OT = USE_OTHR;
B_INSTAL = INSTALLE;
```

```
B_OPER_E = OPER_EAS;
B_MULTIS = MULTI_ST;
B_FACILC = FACIL_CG;
B_SUMCHA = SUM_CHAR;
B_GSC_AD = GSC_ADJU;
B_BASIST = BASIS_TP;
B_WORK_A = WORK_APP;
*;
DATA FPANAL2C;
SET FP.FPANAL2;
IF RATER_ID = 'C';
C_DATA_C = DATA_COM;
C_DIST_F = DIST_FUN;
C_PERFOR = PERFORMA;
C_HEAVYU = HEAVY_US;
C_TRAN_R = TRANS_RT;
C_ONLINE = ONLINE_E;
C_END_US = END_USER;
C_ONLINU = ONLINE_U;
C_COMP_P = COMPLEXP;
C_USE_OT = USE_OTHR;
C_INSTAL = INSTALLE;
C_OPER_E = OPER_EAS;
C_MULTIS = MULTI_ST;
C_FACILC = FACIL_CG;
C_SUMCHA = SUM_CHAR;
C_GSC_AD = GSC_ADJU;
C_BASIST = BASIS_TP;
C_WORK_A = WORK_APP;
*;
DATA FPANAL2D;
SET FP.FPANAL2;
IF RATER_ID = 'D';
D_DATA_C = DATA_COM;
D_DIST_F = DIST_FUN;
D_PERFOR = PERFORMA;
D_HEAVYU = HEAVY_US;
D_TRAN_R = TRANS_RT;
D_ONLINE = ONLINE_E;
D_END_US = END_USER;
D_ONLINU = ONLINE_U;
D_COMP_P = COMPLEXP;
D_USE_OT = USE_OTHR;
D_INSTAL = INSTALLE;
D_OPER_E = OPER_EAS;
D_MULTIS = MULTI_ST;
D_FACILC = FACIL_CG;
D_SUMCHA = SUM_CHAR;
D_GSC_AD = GSC_ADJU;
D_BASIST = BASIS_TP;
D_WORK_A = WORK_APP;
*;
DATA FP.FPANAL2T;              /*  KEY IS SITE/RATER  */;
                               /*  RATER DATA IS IN CLUSTERS OF */;
                               /*  OF COLUMNS  */;
MERGE FPANAL2A FPANAL2B FPANAL2C FPANAL2D;
BY KEY_SA;
```

```
DROP RATER_ID KEY_SR KEY_SRA
     DATA_COM DIST_FUN PERFORMA HEAVY_US
     TRANS_RT ONLINE_E END_USER
     ONLINE_U COMPLEXP USE_OTHR INSTALLE
     OPER_EAS MULTI_ST FACIL_CG SUM_CHAR
     GSC_ADJU WORK_APP;
AVAB_SUM = (A_SUMCHA + B_SUMCHA) / 2;
AVCD_SUM = (C_SUMCHA + D_SUMCHA) / 2;
AVAB_GSC = (A_GSC_AD + B_GSC_AD) / 2;
AVCD_GSC = (C_GSC_AD + D_GSC_AD) / 2;
MRE_ASUM = ABS((AVAB_SUM - A_SUMCHA) / AVAB_SUM);
MRE_CSUM = ABS((AVCD_SUM - C_SUMCHA) / AVCD_SUM);
MRE_AGSC = ABS((AVAB_GSC - A_GSC_AD) / AVAB_GSC);
MRE_BGSC = ABS((AVAB_GSC - B_GSC_AD) / AVAB_GSC);
MRE_CGSC = ABS((AVCD_GSC - C_GSC_AD) / AVCD_GSC);
MRE_DGSC = ABS((AVCD_GSC - D_GSC_AD) / AVCD_GSC);
*;
PROC DELETE DATA = FPANAL2A FPANAL2B
      FPANAL2C FPANAL2D;
*;
*  ==============================================================  */;
*;
PROC SORT DATA = FP.FPANAL1     /*  CREATE FPANAL3  */;
BY KEY_SRA                      /* INCLUDES DATA FROM FPANAL1 & 2 */;
PROC SORT DATA = FP.FPANAL2     /*  KEY IS SITE/RATER/APPLICATION */;
BY KEY_SRA;
*;
DATA FP.FPANAL3;
MERGE FP.FPANAL1 FP.FPANAL2;
BY KEY_SRA;
ADJ_FPTS = GDTOT_FP * GSC_ADJU;
ANALRATE = ADJ_FPTS / TIME_SPT;
*;
*  ==============================================================  */;
*;
DATA FP.APPLICAT                /*  INPUT DATA FROM APPLICAT DAT  */;
INFILE APPLICAT                 /*  KEY IS SITE/APPLICATION#  */;
INPUT
      @1  SITE__NO  2.
      @4  APPLC_NO  1.
      @6  AP_TYPE1  1.
      @8  OTHR_AP1 $1.
      @10 AP_TYPE2  2.
      @12 OTHR_AP2 $1.
      @14 PREV_CNT  4.
      @19 ACT_WHRS  5.;
*;
LENGTH TEMPVAR1 $ 2             /*  ADD MULTI-COLUMN KEY  */;
TEMPVAR1 = SITE__NO;
LENGTH TEMPVAR2 $ 1;
TEMPVAR2 = APPLC_NO;
LENGTH KEY_SA $ 3;
KEY_SA = TEMPVAR1 || TEMPVAR2;
*;
*  ==============================================================  */;
*;
DATA TEMPA/*  CREATE DUPLICATE DATA FOR */;
```

```
SET FP.APPLICAT                 /*  ALL 4 RATERS, SO THERE IS */;
LENGTH RATER_ID $ 1             /*  APPLICATION DATA W/ KEY_SRA */;
RATER_ID = 'A';
KEY_SR = TEMPVAR1 || RATER_ID;
KEY_SRA = KEY_SR || TEMPVAR2;
PROC SORT;
BY KEY_SRA;
*;
DATA TEMPB;
SET FP.APPLICAT;
LENGTH RATER_ID $ 1;
RATER_ID = 'B';
KEY_SR = TEMPVAR1 || RATER_ID;
KEY_SRA = KEY_SR || TEMPVAR2;
PROC SORT;
BY KEY_SRA;
*;
DATA TEMPC;
SET FP.APPLICAT;
LENGTH RATER_ID $ 1;
RATER_ID = 'C';
KEY_SR = TEMPVAR1 || RATER_ID;
KEY_SRA = KEY_SR || TEMPVAR2;
PROC SORT;
BY KEY_SRA;
*;
DATA TEMPD;
SET FP.APPLICAT;
LENGTH RATER_ID $ 1;
RATER_ID = 'D';
KEY_SR = TEMPVAR1 || RATER_ID;
KEY_SRA = KEY_SR || TEMPVAR2;
PROC SORT;
BY KEY_SRA;
*;
DATA FP.APPLIC2                 /* CREATE APPLIC2  */;
MERGE TEMPA TEMPB TEMPC TEMPD   /* KEY IS SITE/RATER/APPLIC#  */;
BY KEY_SRA;
DROP TEMPVAR1 TEMPVAR2;
*;
PROC DELETE DATA = TEMPA TEMPB
   TEMPC TEMPD;
*;
*  ============================================================  */;
*;
PROC SORT DATA = FP.APPLICAT    /*  CREATE FPANAL6  */;
BY KEY_SA                       /*  INCLUDES FPANAL1T, 2T, AND  */;
                                /*  APPLICAT  */;
DATA FP.FPANAL6                 /*  KEY IS SITE/APPLICATION  */;
                                /*  RATER DATA IS IN CLUSTERS  */;
                                /*  OF COLUMNS  */;
MERGE FP.FPANAL1T FP.FPANAL2T FP.APPLICAT;
BY KEY_SA;
ADJ_A_FC = A_FC * A_GSC_AD;
ADJ_B_FC = B_FC * B_GSC_AD;
ADJ_C_FC = C_FC * C_GSC_AD;
ADJ_D_FC = D_FC * D_GSC_AD;
```

```
AVAB_AFC = (ADJ_A_FC + ADJ_B_FC) / 2;
AVCD_AFC = (ADJ_C_FC + ADJ_D_FC) / 2;
AV4_AFC = (ADJ_A_FC + ADJ_B_FC + ADJ_C_FC + ADJ_D_FC) / 4;
MREA_AFC = ABS((AVAB_AFC - ADJ_A_FC) / AVAB_AFC);
MREB_AFC = ABS((AVAB_AFC - ADJ_B_FC) / AVAB_AFC);
MREC_AFC = ABS((AVCD_AFC - ADJ_C_FC) / AVCD_AFC);
MRED_AFC = ABS((AVCD_AFC - ADJ_D_FC) / AVCD_AFC);
MREA4_AC = ABS((AV4_AFC - ADJ_A_FC) / AV4_AFC);
MREB4_AC = ABS((AV4_AFC - ADJ_B_FC) / AV4_AFC);
MREC4_AC = ABS((AV4_AFC - ADJ_C_FC) / AV4_AFC);
MRED4_AC = ABS((AV4_AFC - ADJ_D_FC) / AV4_AFC);
      /* IF ONLY 1 A,B OR C,D USE THE 1 */;
IF ADJ_A_FC = . THEN AVAB_AFC = ADJ_B_FC;
IF ADJ_B_FC = . THEN AVAB_AFC = ADJ_A_FC;
IF ADJ_C_FC = . THEN AVCD_AFC = ADJ_D_FC;
IF ADJ_D_FC = . THEN AVCD_AFC = ADJ_C_FC;
*;
*  ================================================================= */;
*;
DATA TEMPA/*  TEMPORARY DATASETS USED TO  */;
SET FP.FPANAL6                  /*  CREATE MRE_AFPC  */;
IF MREA_AFC NE . ;
RATER_ID = 'A';
MRE_AFPC = MREA_AFC;
MRE4_AC = MREA4_AC;
LENGTH TEMPVAR1 $ 2;
TEMPVAR1 = SITE__NO;
LENGTH KEY_SR $ 3;
KEY_SR = TEMPVAR1 || RATER_ID;
LENGTH TEMPVAR2 $ 1;
TEMPVAR2 = APPLC_NO;
LENGTH KEY_SRA $ 4;
KEY_SRA = KEY_SR || TEMPVAR2;
KEEP KEY_SRA MRE_AFPC MRE4_AC;
PROC SORT DATA = TEMPA;
BY KEY_SRA;
*;
DATA TEMPB;
SET FP.FPANAL6;
IF MREB_AFC NE . ;
RATER_ID = 'B';
MRE_AFPC = MREB_AFC;
MRE4_AC = MREB4_AC;
LENGTH TEMPVAR1 $ 2;
TEMPVAR1 = SITE__NO;
LENGTH KEY_SR $ 3;
KEY_SR = TEMPVAR1 || RATER_ID;
LENGTH TEMPVAR2 $ 1;
TEMPVAR2 = APPLC_NO;
LENGTH KEY_SRA $ 4;
KEY_SRA = KEY_SR || TEMPVAR2;
KEEP KEY_SRA MRE_AFPC MRE4_AC;
PROC SORT DATA = TEMPB;
BY KEY_SRA;
*;
DATA TEMPC;
SET FP.FPANAL6;
```

```
IF MREC_AFC NE . ;
RATER_ID = 'C';
MRE_AFPC = MREC_AFC;
MRE4_AC = MREC4_AC;
LENGTH TEMPVAR1 $ 2;
TEMPVAR1 = SITE__NO;
LENGTH KEY_SR $ 3;
KEY_SR = TEMPVAR1 || RATER_ID;
LENGTH TEMPVAR2 $ 1;
TEMPVAR2 = APPLC_NO;
LENGTH KEY_SRA $ 4;
KEY_SRA = KEY_SR || TEMPVAR2;
KEEP KEY_SRA MRE_AFPC MRE4_AC;
PROC SORT DATA = TEMPC;
BY KEY_SRA;
*;
DATA TEMPD;
SET FP.FPANAL6;
IF MRED_AFC NE . ;
RATER_ID = 'D';
MRE_AFPC = MRED_AFC;
MRE4_AC = MRED4_AC;
LENGTH TEMPVAR1 $ 2;
TEMPVAR1 = SITE__NO;
LENGTH KEY_SR $ 3;
KEY_SR = TEMPVAR1 || RATER_ID;
LENGTH TEMPVAR2 $ 1;
TEMPVAR2 = APPLC_NO;
LENGTH KEY_SRA $ 4;
KEY_SRA = KEY_SR || TEMPVAR2;
KEEP KEY_SRA MRE_AFPC MRE4_AC;
PROC SORT DATA = TEMPD;
BY KEY_SRA;
*;
DATA FP.MREAFPC                /*  CREATE MREAPPC  */;
MERGE TEMPA TEMPB TEMPC TEMPD  /*  INCLUDES MRE'S FOR AFPC'S  */;
BY KEY_SRA                     /*  KEY IS SITE/RATER/APPLICATION */;
*;
PROC DELETE DATA = TEMPA TEMPB TEMPC TEMPD;
*;
*  ===============================================================  */;
*;
DATA FP.RATERBK                /*  INPUT DATA FROM RATER_BK  */;
INFILE RATERBK                 /*  KEY IS SITE/RATER  */;
INPUT
      @1  SITE__NO  2.
      @4  RATER_ID $1.
      @6  APPD_TIM  4.1
      @11 EMPL_TIM  4.1
      @16 ENTITY_R  1.
      @18 FPCT_TIM  4.1
      @23 FO_TRAIN  1.
      @25 TRAN_TIM  4.1
      @30 INTR_CRS  1.
      @32 EXTR_CRS  1.
      @34 SELF_TGT  1.
      @36 EXTERN_1 $3.
```

```
      @40 EXTERN_2 $3.
      @44 OTHR_CRS $1.
      @46 IFPUG_ME  1.
      @48 IF_CONFE  1.;
*;
*  ============================================================= */;
*;
DATA TEMP1                    /*  CREATE DUBLICATE DATA FOR */;
SET FP.RATERBK                /*  APPLICATION #1 AND #2  */;
LENGTH APPLC_CD $ 1           /*  KEY IS SITE/RATER/APPLIC#  */;
APPLC_CD = '1';
LENGTH TEMPVAR1 $ 2           /*  CREATE KEY_SRA  */;
TEMPVAR1 = SITE__NO;
LENGTH KEY_SR $ 3;
KEY_SR = TEMPVAR1 || RATER_ID;
LENGTH KEY_SRA $ 4;
KEY_SRA = KEY_SR || APPLC_CD;
LENGTH KEY_SA $ 3;
KEY_SA = TEMPVAR1 || APPLC_CD;
DROP TEMPVAR1;
PROC SORT;
BY KEY_SRA;
*;
DATA TEMP2;
SET FP.RATERBK;
LENGTH APPLC_CD $ 1;
APPLC_CD = '2';
LENGTH TEMPVAR1 $ 2;
TEMPVAR1 = SITE__NO;
LENGTH KEY_SR $ 3;
KEY_SR = TEMPVAR1 || RATER_ID;
LENGTH KEY_SRA $ 4;
KEY_SRA = KEY_SR || APPLC_CD;
LENGTH KEY_SA $ 3;
KEY_SA = TEMPVAR1 || APPLC_CD;
DROP TEMPVAR1;
PROC SORT;
BY KEY_SRA;
*;
DATA FP.RATERBK2;
MERGE TEMP1 TEMP2;
BY KEY_SRA;
*;
PROC DELETE DATA = TEMP1 TEMP2;
*;
*  ============================================================  */;
*;
DATA RATERBKA                 /*  CREATE TEMPORARY DATASETS  */;
SET FP.RATERBK2               /*  USED TO CREATE RATERBKT  */;
IF RATER_ID = 'A';
A_APPD_T = APPD_TIM;
A_EMPL_T = EMPL_TIM;
A_ENTITY = ENTITY_R;
A_FPCT_T = FPCT_TIM;
A_FO_TRA = FO_TRAIN;
A_TRAN_T = TRAN_TIM;
A_IFPUGM = IFPUG_ME;
```

```
*;
DATA RATERBKB;
SET FP.RATERBK2;
IF RATER_ID = 'B';
B_APPD_T = APPD_TIM;
B_EMPL_T = EMPL_TIM;
B_ENTITY = ENTITY_R;
B_FPCT_T = FPCT_TIM;
B_FO_TRA = FO_TRAIN;
B_TRAN_T = TRAN_TIM;
B_IFPUGM = IFPUG_ME;
*;
DATA RATERBKC;
SET FP.RATERBK2;
IF RATER_ID = 'C';
C_APPD_T = APPD_TIM;
C_EMPL_T = EMPL_TIM;
C_ENTITY = ENTITY_R;
C_FPCT_T = FPCT_TIM;
C_FO_TRA = FO_TRAIN;
C_TRAN_T = TRAN_TIM;
C_IFPUGM = IFPUG_ME;
*;
DATA RATERBKD;
SET FP.RATERBK2;
IF RATER_ID = 'D';
D_APPD_T = APPD_TIM;
D_EMPL_T = EMPL_TIM;
D_ENTITY = ENTITY_R;
D_FPCT_T = FPCT_TIM;
D_FO_TRA = FO_TRAIN;
D_TRAN_T = TRAN_TIM;
D_IFPUGM = IFPUG_ME;
*;
DATA FP.RATERBKT;
MERGE RATERBKA RATERBKB RATERBKC RATERBKD;
BY KEY_SA;
DROP RATER_ID EMPL_TIM ENTITY_R FPCT_TIM
     FO_TRAIN TRAN_TIM INTR_CRS EXTR_CRS
     SELF_TGT EXTERN_1 EXTERN_2 OTHR_CRS
     IFPUG_ME IF_CONFE;
*;
PROC DELETE DATA = RATERBKA RATERBKB
   RATERBKC RATERBKD;
*;
*  ============================================================  */;
*;
DATA FP.FPANAL4                /*  CREATE FPANAL4 - A MERGE OF  */;
MERGE FP.FPANAL6 FP.RATERBKT   /*  FPANAL6 & RATERBKT  */;
BY KEY_SA;
*;
*  ============================================================  */;
*;
DATA FP.LANGUAGE               /*  INPUT DATA FROM LANGUAGE DAT  */;
INFILE LANGUAGE;
INPUT
      @1  SITE__NO  2.
```

```
      @4- APPLC_NO  1.
      @6  LANGUAG1 $1.
      @8  LNS_COD1  7.
      @16 LANGUAG2 $1.
      @18 LNS_COD2  7.
      @26 LANGUAG3 $1.
      @28 LNS_COD3  7.
      @36 LANGUAG4 $1.
      @38 LNS_COD4  7.
      @46 OTHR_LAN $1;
*;
*  ================================================================  */;
*;
DATA FP.MICROCA                /*  INPUT DATA FROM MICRO_CA DAT  */;
INFILE MICROCA;
INPUT
      @1  SITE__NO  2.
      @4  BACKUP_F  1.
      @6  OTHR_BAC $1.
      @8  MULTI_FU  1.
      @10 OTHR_MUL $1.
      @12 ERROR_MS  1.
      @14 OTHR_ERR $1.
      @16 MENU_FTS  1.
      @18 MENU_NOF  1.
      @20 OTHR_MEN $1.
      @22 HELP_MNF  1.
      @24 OTHR_HP1 $1.
      @26 HELP_MFT  1.
      @28 OTHR_HP2 $1.
      @30 HELP_SNF  1.
      @32 OTHR_HP3 $1.
      @34 HELP_SFT  1.
      @36 OTHR_HP4 $1.
      @38 SUBTOTAL  1.
      @40 OTHR_SUB $1.
      @42 HARD_TBL  1.
      @44 OTHR_TBL $1.
      @46 SELECTCT  1.
      @48 OTHR_SEL $1.
      @50 ORDER_CT  1.
      @52 OTHR_ORD $1.
      @54 INQU_WTS  1.
      @56 OTHR_INQ $1.
      @58 EXT_FILE  1.
      @60 OTHR_EX1 $1.
      @62 TRANS_FL  1.
      @64 OTHR_EX2 $1.;
*;
*  ================================================================  */;
*;
DATA FP.SITEQUS                /*  INPUT DATA FROM SITEQUS DAT  */;
INFILE SITEQUS;
INPUT
      @1  SITE__NO  2.
      @4  INDUSTRY  2.
      @7  SITE_TIM  4.1
```

```
      @12 METHODOL  1.
      @14 OTHR_MET $1.
      @16 AUTOMATE  1.
      @18 AUTO_NAM  1.
      @20 OTHR_AUT $1.
      @22 TRAINING  1.
      @24 TRAIN_HR  3.
      @28 DOCUMENT  1.
      @30 DOC_PAGE  4.
      @35 OTHR_DOC $1.
      @37 NEW_MANP  1.
      @39 NEW_CHGS  1.
      @41 NEW_PROG  1.
      @43 NEW_PROD  1.
      @45 NEW_RPTG  1.
      @47 NEW_OTHR  1.
      @49 OTHR_NEW $1.
      @51 MAI_MANP  1.
      @53 MAI_CHGS  1.
      @55 MAI_PROG  1.
      @57 MAI_PROD  1.
      @59 MAI_RPTG  1.
      @61 MAI_OTHR  1.
      @63 OTHR_MAI $1.
      @65 CONTINGE  1.
      @67 CONT_PER  3.
      @71 OTHR_CMT $1.;
*;
*  ==========================================================  */;
*;
DATA THEWORKS;
MERGE FP.FPANAL3  FP.APPLIC2
      FP.RATERBK2 FP.MREAFPC;
BY KEY_SRA;
*;
                               /* TEMPORARY DATASETS FOR  */;
DATA TEMP1                     /* CREATING FP.WORKS2 WITH RATERS */;
SET THEWORKS                   /* X & Y AND METHODS I & E  */;
IF RATER_ID = 'A'              /* FROM FPANAL1 */;
LENGTH RATER_CD $ 1;
RATER_CD = 'X';
LENGTH METHD_CD $ 1;
METHD_CD = 'I';
LENGTH TEMPVAR1 $ 2;
TEMPVAR1 = SITE__NO;
LENGTH TEMPVAR2 $ 1;
TEMPVAR2 = APPLC_NO;
KEY_SA = TEMPVAR1 || TEMPVAR2;
KEY_SAM = KEY_SA || METHD_CD;
X_UFPC = GDTOT_FP              /* FROM FPANAL3 */;
X_AFPT = ADJ_FPTS              /* FROM FPANAL3 */;
X_BASIS = BASIS_TP             /* FROM FPANAL3 */;
X_WORKAP = WORK_APP            /* FROM FPANAL3 */;
X_ANALRT = ANALRATE            /* FROM FPANAL3 */;
X_APTYP1 = AP_TYPE1            /* FROM APPLIC3 */;
X_APTYP2 = AP_TYPE2            /* FROM APPLIC3 */;
X_APPDTM = APPD_TIM            /* FROM APPLIC3 */;
```

```
X_FPCTTM = FPCT_TIM             /* FROM RATERBK2 */;
X_TRAING = FO_TRAIN             /* FROM RATERBK2 */;
X_MRE_AC = MRE_AFPC             /* FROM MREAFPC */;
X_MRE4AC = MRE4_AC              /* FROM MREAFPC */;
KEEP KEY_SAM  X_AFPT   X_BASIS  X_WORKAP
     X_ANALRT X_APTYP1 X_APTYP2 X_APPDTM
     X_FPCTTM X_TRAING X_MRE_AC X_MRE4AC
     RATER_ID;
PROC SORT;
BY KEY_SAM;
*;
DATA TEMP2;
SET THEWORKS;
IF RATER_ID = 'B';
LENGTH RATER_CD $ 1;
RATER_CD = 'Y';
LENGTH METHD_CD $ 1;
METHD_CD = 'I';
LENGTH TEMPVAR1 $ 2;
TEMPVAR1 = SITE__NO;
LENGTH TEMPVAR2 $ 1;
TEMPVAR2 = APPLC_NO;
KEY_SA = TEMPVAR1 || TEMPVAR2;
KEY_SAM = KEY_SA || METHD_CD;
Y_AFPT = ADJ_FPTS;
Y_BASIS = BASIS_TP;
Y_WORKAP = WORK_APP;
Y_ANALRT = ANALRATE;
Y_APTYP1 = AP_TYPE1;
Y_APTYP2 = AP_TYPE2;
Y_APPDTM = APPD_TIM;
Y_FPCTTM = FPCT_TIM;
Y_TRAING = FO_TRAIN;
Y_MRE_AC = MRE_AFPC;
Y_MRE4AC = MRE4_AC;
KEEP KEY_SAM  Y_AFPT   Y_BASIS  Y_WORKAP
     Y_ANALRT Y_APTYP1 Y_APTYP2 Y_APPDTM
     Y_FPCTTM Y_TRAING Y_MRE_AC
     Y_MRE4AC RATER_ID;
PROC SORT;
BY KEY_SAM;
*;
DATA TEMP3;
SET THEWORKS;
IF RATER_ID = 'C';
LENGTH RATER_CD $ 1;
RATER_CD = 'X';
LENGTH METHD_CD $ 1;
METHD_CD = 'E';
LENGTH TEMPVAR1 $ 2;
TEMPVAR1 = SITE__NO;
LENGTH TEMPVAR2 $ 1;
TEMPVAR2 = APPLC_NO;
KEY_SA = TEMPVAR1 || TEMPVAR2;
KEY_SAM = KEY_SA || METHD_CD;
X_AFPT = ADJ_FPTS;
X_BASIS = BASIS_TP;
```

```
X_WORKAP = WORK_APP;
X_ANALRT = ANALRATE;
X_APTYP1 = AP_TYPE1;
X_APTYP2 = AP_TYPE2;
X_APPDTM = APPD_TIM;
X_FPCTTM = FPCT_TIM;
X_TRAING = FO_TRAIN;
X_MRE_AC = MRE_AFPC;
X_MRE4AC = MRE4_AC;
KEEP KEY_SAM  X_AFPT   X_BASIS  X_WORKAP
     X_ANALRT X_APTYP1 X_APTYP2 X_APPDTM
     X_FPCTTM X_TRAING X_MRE_AC
     X_MRE4AC RATER_ID;
PROC SORT;
BY KEY_SAM;
*;
DATA TEMP4;
SET THEWORKS;
IF RATER_ID = 'D';
LENGTH RATER_CD $ 1;
RATER_CD = 'Y';
LENGTH METHD_CD $ 1;
METHD_CD = 'E';
LENGTH TEMPVAR1 $ 2;
TEMPVAR1 = SITE__NO;
LENGTH TEMPVAR2 $ 1;
TEMPVAR2 = APPLC_NO;
KEY_SA = TEMPVAR1 || TEMPVAR2;
KEY_SAM = KEY_SA || METHD_CD;
Y_AFPT = ADJ_FPTS;
Y_BASIS = BASIS_TP;
Y_WORKAP = WORK_APP;
Y_ANALRT = ANALRATE;
Y_APTYP1 = AP_TYPE1;
Y_APTYP2 = AP_TYPE2;
Y_APPDTM = APPD_TIM;
Y_FPCTTM = FPCT_TIM;
Y_TRAING = FO_TRAIN;
Y_MRE_AC = MRE_AFPC;
Y_MRE4AC = MRE4_AC;
KEEP KEY_SAM  Y_AFPT   Y_BASIS  Y_WORKAP
     Y_ANALRT Y_APTYP1 Y_APTYP2 Y_APPDTM
     Y_FPCTTM Y_TRAING Y_MRE_AC
     Y_MRE4AC RATER_ID;
PROC SORT;
BY KEY_SAM;
*;
DATA FP.WORKS2;
MERGE TEMP1 TEMP3 TEMP2 TEMP4;
BY KEY_SAM;
*;
PROC DELETE DATA = TEMP1 TEMP2 TEMP3 TEMP4;
*;
PROC PRINT;
VAR KEY_SAM  X_AFPT   Y_AFPT   X_BASIS  X_WORKAP
    X_ANALRT X_APTYP1 X_APTYP2 X_APPDTM
    X_APPDTM X_FPCTTM X_TRAING X_MRE_AC Y_MRE_AC
```

```
    X_MRE4AC Y_MRE4AC;
*;
                                  /*  THE END  */;
```

This program, ANALYZE.SAS, analyzes inter-rater and inter-method reliability.

```
*  ANALYZE INTER-RATER RELIABILITY FOR ALL LEVELS  */;
*;
PROC CORR PEARSON SPEARMAN/* CORR COEFS FOR # INPUT F TYPES */
  DATA = FP.FPANAL1T;
VAR A_INPU_L B_INPU_L C_INPU_L D_INPU_L;
PROC CORR PEARSON SPEARMAN DATA = FP.FPANAL1T;
VAR A_INPU_M B_INPU_M C_INPU_M D_INPU_M;
PROC CORR PEARSON SPEARMAN DATA = FP.FPANAL1T;
VAR A_INPU_H B_INPU_H C_INPU_H D_INPU_H;
*;
PROC CORR PEARSON SPEARMAN/* CORR COEFS FOR # OUTPUT F TYPES */
  DATA = FP.FPANAL1T;
VAR A_OUTP_L B_OUTP_L C_OUTP_L D_OUTP_L;
PROC CORR PEARSON SPEARMAN DATA = FP.FPANAL1T;
VAR A_OUTP_M B_OUTP_M C_OUTP_M D_OUTP_M;
PROC CORR PEARSON SPEARMAN DATA = FP.FPANAL1T;
VAR A_OUTP_H B_OUTP_H C_OUTP_H D_OUTP_H;
*;
PROC CORR PEARSON SPEARMAN     /* CORR COEFS FOR # FILES F TYPES */
  DATA = FP.FPANAL1T;
VAR A_FILE_L B_FILE_L C_FILE_L D_FILE_L;
PROC CORR PEARSON SPEARMAN DATA = FP.FPANAL1T;
VAR A_FILE_M B_FILE_M C_FILE_M D_FILE_M;
PROC CORR PEARSON SPEARMAN DATA = FP.FPANAL1T;
VAR A_FILE_H B_FILE_H C_FILE_H D_FILE_H;
*;
PROC CORR PEARSON SPEARMAN     /* CORR COEFS FOR # INQUIR F TYPES */
  DATA = FP.FPANAL1T;
VAR A_INQU_L B_INQU_L C_INQU_L D_INQU_L;
PROC CORR PEARSON SPEARMAN DATA = FP.FPANAL1T;
VAR A_INQU_M B_INQU_M C_INQU_M D_INQU_M;
PROC CORR PEARSON SPEARMAN DATA = FP.FPANAL1T;
VAR A_INQU_H B_INQU_H C_INQU_H D_INQU_H;
*;
PROC CORR PEARSON SPEARMAN     /* CORR COEFS FOR # INTERF F TYPES */
  DATA = FP.FPANAL1T;
VAR A_INTR_L B_INTR_L C_INTR_L D_INTR_L;
PROC CORR PEARSON SPEARMAN DATA = FP.FPANAL1T;
VAR A_INTR_M B_INTR_M C_INQU_M D_INTR_M;
PROC CORR PEARSON SPEARMAN DATA = FP.FPANAL1T;
VAR A_INTR_H B_INTR_H C_INTR_H D_INTR_H;
*;
PROC PLOT DATA = FP.FPANAL1T    /*  ANALYZE FUNC COUNTS FOR INPUTS */;
    PLOT A_TO_INP * B_TO_INP;
    TITLE 'Rater A v. Rater B Inputs';
PROC PLOT DATA = FP.FPANAL1T;
    PLOT C_TO_INP * D_TO_INP;
    TITLE 'Rater C v. Rater D Inputs';
PROC CORR SPEARMAN DATA = FP.FPANAL1T;
VAR A_TO_INP B_TO_INP C_TO_INP D_TO_INP AVAB_INP AVCD_INP;
*;
PROC PLOT DATA = FP.FPANAL1T    /*  ANALYZE FUNC COUNTS FOR OUTPUT */;
    PLOT A_TO_OUT * B_TO_OUT;
    TITLE 'Rater A v. Rater B Outputs';
```

```
PROC PLOT DATA = FP.FPANAL1T;
    PLOT C_TO_OUT * D_TO_OUT;
    TITLE 'Rater C v. Rater D Outputs';
PROC CORR SPEARMAN DATA = FP.FPANAL1T;
VAR A_TO_OUT B_TO_OUT C_TO_OUT D_TO_OUT AVAB_OUT AVCD_OUT;
*;
PROC PLOT DATA = FP.FPANAL1T    /*  ANALYZE FUNC COUNTS FOR FILES */;
    PLOT A_TO_FIL * B_TO_FIL;
    TITLE 'Rater A v. Rater B Files';
PROC PLOT DATA = FP.FPANAL1T;
    PLOT C_TO_FIL * D_TO_FIL;
    TITLE 'Rater C v. Rater D Files';
PROC CORR SPEARMAN DATA = FP.FPANAL1T;
VAR A_TO_FIL B_TO_FIL C_TO_FIL D_TO_FIL AVAB_FIL AVCD_FIL;
*;
PROC PLOT DATA = FP.FPANAL1T    /*  ANALYZE FUNC CNTS FOR INQUIRIE */;
    PLOT A_TO_INQ * B_TO_INQ;
    TITLE 'Rater A v. Rater B Inquiries';
PROC PLOT DATA = FP.FPANAL1T;
    PLOT C_TO_INQ * D_TO_INQ;
    TITLE 'Rater C v. Rater D Inquiries';
PROC CORR SPEARMAN DATA = FP.FPANAL1T;
VAR A_TO_INQ B_TO_INQ C_TO_INQ D_TO_INQ AVAB_INQ AVCD_INQ;
*;
PROC PLOT DATA = FP.FPANAL1T    /*  ANALYZE FUNC CNTS FOR INTERFAC */;
    PLOT A_TO_INT * B_TO_INT;
    TITLE 'Rater A v. Rater B Interface Files';
PROC PLOT DATA = FP.FPANAL1T;
    PLOT C_TO_INT * D_TO_INT;
    TITLE 'Rater C v. Rater D Interface Files';
PROC CORR SPEARMAN DATA = FP.FPANAL1T;
VAR A_TO_INT B_TO_INT C_TO_INT D_TO_INT AVAB_INT AVCD_INT;
*;
PROC PLOT DATA = FP.FPANAL1T    /*  ANALYZE A-TOTAL V. B-TOTAL  */;
    PLOT A_FC * B_FC;
    TITLE 'Rater A v. Rater B. Unadjusted Function Point Count';
PROC CORR SPEARMAN DATA = FP.FPANAL1T;
VAR A_FC B_FC;
*;
PROC PLOT DATA = FP.FPANAL1T    /*  ANALYZE C-TOTAL V. D-TOTAL  */;
    PLOT C_FC * D_FC;
    TITLE 'Rater C v. Rater D Unadjusted Function Point Count';
PROC CORR SPEARMAN DATA = FP.FPANAL1T;
VAR C_FC D_FC;
*;
PROC MEANS DATA = FP.FPANAL1T   /*  CALCULATE AVG MRE'S  */;
VAR MRE_AINP MRE_CINP MRE_AOUT MRE_COUT
    MRE_AFIL MRE_CFIL MRE_AINQ MRE_CINQ
    MRE_AINT MRE_CINT
    MRE_A_FC MRE_C_FC;
*;
PROC CORR PEARSON SPEARMAN/*  DO CORR COEFF'S ON GSC'S  */
  DATA = FP.FPANAL2T;
VAR A_DATA_C B_DATA_C C_DATA_C D_DATA_C;
PROC CORR PEARSON SPEARMAN DATA = FP.FPANAL2T;
VAR A_DIST_F B_DIST_F C_DIST_F D_DIST_F;
PROC CORR PEARSON SPEARMAN DATA = FP.FPANAL2T;
```

```
VAR A_PERFOR B_PERFOR C_PERFOR D_PERFOR;
PROC CORR PEARSON SPEARMAN DATA = FP.FPANAL2T;
VAR A_HEAVYU B_HEAVYU C_HEAVYU D_HEAVYU;
PROC CORR PEARSON SPEARMAN DATA = FP.FPANAL2T;
VAR A_TRAN_R B_TRAN_R C_TRAN_R D_TRAN_R;
PROC CORR PEARSON SPEARMAN DATA = FP.FPANAL2T;
VAR A_ONLINE B_ONLINE C_ONLINE D_ONLINE;
PROC CORR PEARSON SPEARMAN DATA = FP.FPANAL2T;
VAR A_END_US B_END_US C_END_US D_END_US;
PROC CORR PEARSON SPEARMAN DATA = FP.FPANAL2T;
VAR A_ONLINU B_ONLINU C_ONLINU D_ONLINU;
PROC CORR PEARSON SPEARMAN DATA = FP.FPANAL2T;
VAR A_COMP_P B_COMP_P C_COMP_P D_COMP_P;
PROC CORR PEARSON SPEARMAN DATA = FP.FPANAL2T;
VAR A_USE_OT B_USE_OT C_USE_OT D_USE_OT;
PROC CORR PEARSON SPEARMAN DATA = FP.FPANAL2T;
VAR A_INSTAL B_INSTAL C_INSTAL D_INSTAL;
PROC CORR PEARSON SPEARMAN DATA = FP.FPANAL2T;
VAR A_OPER_E B_OPER_E C_OPER_E D_OPER_E;
PROC CORR PEARSON SPEARMAN DATA = FP.FPANAL2T;
VAR A_MULTIS B_MULTIS C_MULTIS D_MULTIS;
PROC CORR PEARSON SPEARMAN DATA = FP.FPANAL2T;
VAR A_FACILC B_FACILC C_FACILC D_FACILC;
*;
PROC PLOT DATA = FP.FPANAL2T   /*  ANALYZE GSC ADJUSTMENT  */;
    PLOT A_GSC_AD * B_GSC_AD;
    TITLE 'Rater A v. Rater B General System Characteristics';
PROC PLOT DATA = FP.FPANAL2T;
    PLOT C_GSC_AD * D_GSC_AD;
    TITLE 'Rater C v. Rater D General System Characteristics';
PROC CORR PEARSON SPEARMAN DATA = FP.FPANAL2T;
VAR A_SUMCHA B_SUMCHA C_SUMCHA D_SUMCHA;
PROC CORR PEARSON SPEARMAN DATA = FP.FPANAL2T;
VAR A_GSC_AD B_GSC_AD C_GSC_AD D_GSC_AD;
PROC MEANS DATA = FP.FPANAL2T;
VAR MRE_ASUM MRE_CSUM MRE_AGSC MRE_CGSC;
*;
PROC MEANS DATA = FP.FPANAL6   /*  MRE'S FOR AFPC'S  */;
VAR MREA_AFC MREC_AFC
    MREA4_AC MREB4_AC MREC4_AC MRED4_AC;
                               /*  AFPC CORR COEFFS  */;
PROC CORR PEARSON SPEARMAN DATA = FP.FPANAL6;
VAR ADJ_A_FC ADJ_B_FC ADJ_C_FC ADJ_D_FC;
*;
*  ============================================================== */;
*  ANALYZE INTER-METHOD RELIABILITY  */;
*;
PROC PLOT DATA = FP.FPANAL1T    /*  PLOTS M1 V. M2 FOR FUNCTION  */;
    PLOT AVAB_INP * AVCD_INP    /*  TYPES  */;
    TITLE 'IFPUG v. E-R Method for Inputs';
PROC PLOT DATA = FP.FPANAL1T;
    PLOT AVAB_FIL * AVCD_FIL;
PROC PLOT DATA = FP.FPANAL1T;
    PLOT AVAB_OUT * AVCD_OUT;
    TITLE 'IFPUG v. E-R Method for Outputs';
PROC PLOT DATA = FP.FPANAL1T;
    PLOT AVAB_INQ * AVCD_INQ;
```

```
    TITLE 'IFPUG v. E-R Method for Inquiries';
PROC PLOT DATA = FP.FPANAL1T;
    PLOT AVAB_INT * AVCD_INT;
    TITLE 'IFPUG v. E-R Method for Interface Files';
*;
PROC PLOT DATA = FP.FPANAL1T    /*  ANALYZE M1 V. M2 UFPC  */;
    PLOT AVGAB_FC * AVGCD_FC;
    TITLE 'IFPUG v. E-R Method for Undajusted Function Pt Counts';
*;
PROC CORR PEARSON SPEARMAN DATA = FP.FPANAL1T;
VAR AVGAB_FC AVGCD_FC;
*;
PROC REG DATA = FP.FPANAL1T     /*  METHOD 1 V. 2 REGRESSIONS  */;
     TITLE 'Method 1 v. Method 2 for Inputs';
     MODEL AVAB_INP = AVCD_INP / NOINT;
PROC REG DATA = FP.FPANAL1T;
     TITLE 'Method 1 v. Method 2 for Outputs';
     MODEL AVAB_OUT = AVCD_OUT / NOINT;
PROC REG DATA = FP.FPANAL1T;
     TITLE 'Method 1 v. Method 2 for Files';
     MODEL AVAB_FIL = AVCD_FIL / NOINT;
PROC REG DATA = FP.FPANAL1T;
     TITLE 'Method 1 v. Method 2 for Inquiries';
     MODEL AVAB_INQ = AVCD_INQ / NOINT;
PROC REG DATA = FP.FPANAL1T;
     TITLE 'Method 1 v. Method 2 for Interfaces';
     MODEL AVAB_INT = AVCD_INT / NOINT;
*;
PROC REG DATA = FP.FPANAL1T;
TITLE 'IFPUG v. E-R Method 2 for UFPC (intercept = 0)';
MODEL AVGAB_FC = AVGCD_FC /P R COLLIN DW NOINT;
OUTPUT OUT=TEMP1 R=RESID1 P=YHAT1;
PROC PLOT DATA=TEMP1;
PLOT RESID1*YHAT1;
*;
PROC REG DATA = FP.FPANAL1T;
TITLE 'IFPUG v. E-R Method for UFPC';
MODEL AVGAB_FC = AVGCD_FC /P R COLLIN DW;
*;
PROC CORR PEARSON SPEARMAN/*CORR COEFS FOR ADJ FROM 4 + PREV*/
  DATA = FP.FPANAL6;
VAR ADJ_A_FC ADJ_B_FC ADJ_C_FC ADJ_D_FC PREV_CNT;
*;
                              /*  THE END;
```

This program tabulates the responses to questions about the raters, the applications, the sites, and the microcases.

```
CMS FILEDEF OUTPUT1 DISK OUTPUT1 DATA A;
CMS FILEDEF OUTPUT2 DISK OUTPUT2 DATA A;
CMS FILEDEF OUTPUT3 DISK OUTPUT3 DATA A;
*;
*  ============================================================== */;
*  COMPARE METHOD 1 V. METHOD 2 FACTORS */;

PROC MEANS DATA = FP.RATERBKT   /*  EXPERIENCE LEVELS */;
VAR A_APPD_T B_APPD_T C_APPD_T D_APPD_T
    A_FPCT_T B_FPCT_T C_FPCT_T D_FPCT_T;
*;
* ==============================================================  */;
*;
PROC FREQ DATA = FP.RATERBKT    /* TABULATE E-R EXPERIENCE */;
TABLES C_ENTITY D_ENTITY;
*;
*  ==============================================================  */;
*;
PROC FREQ DATA = FP.APPLICAT    /*  TABULATE APPLICATION TYPES  */;
TABLES AP_TYPE1 AP_TYPE2;
*;
*  ==============================================================  */;
*  TABULATE RATER BACKGROUNDS  */;
*;
PROC FREQ DATA = FP.RATERBK;
TABLES APPD_TIM EMPL_TIM ENTITY_R
       FPCT_TIM FO_TRAIN TRAN_TIM
       INTR_CRS EXTR_CRS SELF_TGT
       EXTERN_1 EXTERN_2 IFPUG_ME
       IF_CONFE;
PROC MEANS DATA = FP.RATERBK;
VAR APPD_TIM EMPL_TIM FPCT_TIM TRAN_TIM;
*;
*  ==============================================================  */;
*  TABULATE DATA FROM MICRO CASES  */;
*;
PROC FREQ DATA = FP.MICROCA;
TABLES BACKUP_F MULTI_FU ERROR_MS MENU_FTS MENU_NOF HELP_MNF HELP_MFT
       HELP_SNF HELP_SFT SUBTOTAL HARD_TBL SELECTCT ORDER_CT INQU_WTS
       EXT_FILE TRANS_FL;
*;
*  ==============================================================  */;
*  TABULATE DATA FROM SITE QUESTIONS  */;
*;
PROC FREQ DATA = FP.SITEQUS;
TABLES INDUSTRY METHODOL AUTOMATE AUTO_NAM
       TRAINING TRAIN_HR DOCUMENT
       NEW_MANP NEW_CHGS NEW_PROG
  NEW_PROD NEW_RPTG NEW_OTHR
       MAI_MANP MAI_CHGS MAI_PROG
       MAI_PROD MAI_RPTG MAI_OTHR
       CONTINGE CONT_PER;
PROC MEANS DATA = FP.SITEQUS;
VAR SITE_TIM TRAIN_HR DOC_PAGE CONT_PER;
```

```
*;
*  ============================================================  */;
*  ODDS AND ENDS  */;
*;
PROC MEANS DATA = FP.FPANAL2    /*  AVG GSC AND AVG TIME_SPENT  */;
VAR SUM_CHAR GSC_ADJU TIME_SPT;
*;
PROC FREQ DATA = FP.FPANAL2     /*  TABULATE GSC & ANALYSIS DATA  */;
TABLES SUM_CHAR GSC_ADJU TIME_SPT BASIS_TP WORK_APP;
*;
PROC MEANS DATA = FP.FPANAL3   /*  STATISTICS ON FP COUNTING RATE */;
VAR ANALRATE ADJ_FPTS          /*  AND APPLICATION SIZE  */;
*;
PROC FREQ DATA = FP.FPANAL3;
TABLES ANALRATE ADJ_FPTS;
*;
DATA TEMP1;
SET FP.FPANAL6;
EFORT_AB = ACT_WHRS / AVAB_AFC;
EFORT_4 = ACT_WHRS / AV4_AFC;
PROC MEANS;
VAR EFORT_AB EFORT_4 AVAB_AFC AV4_AFC;
*;
*  ============================================================  */;
*  OUTPUT TABLES OF DATA;
*;
DATA _NULL_                    /*  OUTPUT DATA FROM FPANAL1T */;
SET FP.FPANAL1T                /*  IS DATA FOR PLOTS  */;
FILE OUTPUT2;
PUT
      @1  SITE__NO  2.
      @4  APPLC_NO  1.
      @6  KEY_SA   $3.
      @10 A_FC      4.
      @15 B_FC      4.
      @20 C_FC      4.
      @25 D_FC      4.
      @30 AVGAB_FC  4.
      @35 AVGCD_FC  4.
      @40 MRE_A_FC  5.2
      @46 MRE_B_FC  5.2
      @52 MRE_C_FC  5.2
      @58 MRE_D_FC  5.2;
*;
DATA _NULL_                    /*  OUTPUT DATA FROM FPANAL 1&2  */;
SET FP.FPANAL3                 /*  IS AN OVERALL CHECK OF DATA  */;
FILE OUTPUT3;
PUT
      @1  SITE__NO  2.
      @4  RATER_ID $1.
      @6  APPLC_NO  1.
      @8  GDTOT_FP  5.
      @14 GSC_ADJU  6.3
      @21 ADJ_FPTS  5.;
*;
      /*  THE END  */;
```

This program analyzes the effects of experience, application size, etc. on reliability.

```
*  TABULATE EXPERIENCE LEVELS   */;
*;
PROC MEANS DATA = FP.RATERBKT;
VAR A_APPD_T B_APPD_T C_APPD_T D_APPD_T
    A_FPCT_T B_FPCT_T C_FPCT_T D_FPCT_T;
PROC FREQ DATA = FP.RATERBKT;
TABLES C_ENTITY D_ENTITY;
*;
*  ================================================================  */;
*  OVERALL CORRELATION COEFFICIENTS AND MRE'S  */;
*;
PROC CORR SPEARMAN DATA = FP.WORKS2;
VAR X_AFPT Y_AFPT;
PROC MEANS DATA = FP.WORKS2;
VAR X_MRE_AC Y_MRE_AC X_MRE4AC Y_MRE4AC;
*;
*  ================================================================  */;
*  CALCULATE CORRELATION COEFFICIENTS BY PROJECT BASIS TYPE  */;
*;
DATA BASIS1;
SET FP.WORKS2;
IF (X_BASIS = 1 OR X_BASIS = 4) AND
   (Y_BASIS = 1 OR Y_BASIS = 4);
PROC CORR SPEARMAN;
VAR X_AFPT Y_AFPT;
TITLE 'Analyses Based on Requirements Def.';
PROC MEANS;
VAR X_MRE_AC Y_MRE_AC X_MRE4AC Y_MRE4AC;
*;
DATA BASIS2;
SET FP.WORKS2;
IF (X_BASIS = 2 OR X_BASIS = 5) AND
   (Y_BASIS = 2 OR Y_BASIS = 5);
PROC CORR SPEARMAN;
VAR X_AFPT Y_AFPT;
TITLE 'Analyses Based on Detailed Design';
PROC MEANS;
VAR X_MRE_AC Y_MRE_AC X_MRE4AC Y_MRE4AC;
*;
DATA BASIS3;
SET FP.WORKS2;
IF X_BASIS = 3 AND Y_BASIS = 3;
PROC CORR SPEARMAN;
VAR X_AFPT Y_AFPT;
TITLE 'Analyses Based on Implemented Systems';
PROC MEANS;
VAR X_MRE_AC Y_MRE_AC X_MRE4AC Y_MRE4AC;
*;
DATA BASIS4;
SET FP.WORKS2;
IF ((X_BASIS = 1 OR X_BASIS= 4) AND (Y_BASIS NE 1 AND Y_BASIS NE 4)) OR
   ((X_BASIS = 2 OR X_BASIS= 5) AND (Y_BASIS NE 2 AND Y_BASIS NE 5)) OR
    (X_BASIS = 3 AND Y_BASIS NE 3);
```

```
PROC CORR SPEARMAN;
VAR X_AFPT Y_AFPT;
TITLE 'Analyses Where Raters had Mixed Bases';
PROC MEANS;
VAR X_MRE_AC Y_MRE_AC X_MRE4AC Y_MRE4AC;
*;
PROC DELETE DATA = BASIS1 BASIS2 BASIS3;
*;
*  ============================================================  */;
*  CALCULATE CORRELATION COEFFICIENTS BY FAMILIARITY W/ APPLICATION */;
*;
DATA FAMILRY;
SET FP.WORKS2;
IF (X_WORKAP = 1 OR X_WORKAP = 2) AND
   (Y_WORKAP = 1 OR Y_WORKAP = 2);
PROC CORR SPEARMAN;
VAR X_AFPT Y_AFPT;
TITLE 'Analyses by Raters Who Worked on System';
PROC MEANS;
VAR X_MRE_AC Y_MRE_AC X_MRE4AC Y_MRE4AC;
*;
DATA FAMILRN;
SET FP.WORKS2;
IF X_WORKAP = 0 AND Y_WORKAP = 0;
PROC CORR SPEARMAN;
VAR X_AFPT Y_AFPT;
TITLE 'Analyses by Raters Who Did Not Work on System';
PROC MEANS;
VAR X_MRE_AC Y_MRE_AC X_MRE4AC Y_MRE4AC;
*;
DATA FAMILRMX;
SET FP.WORKS2;
IF (X_WORKAP = 0 AND Y_WORKAP NE 0) OR
   (X_WORKAP NE 0 AND Y_WORKAP = 0);
PROC CORR SPEARMAN;
VAR X_AFPT Y_AFPT;
TITLE 'Analyses w/ Mixed Raters (Worked & Not Worked on System)';
PROC MEANS;
VAR X_MRE_AC Y_MRE_AC X_MRE4AC Y_MRE4AC;
*;
PROC DELETE DATA = FAMILRY FAMILRN FAMILRMX;
*;
*  ============================================================= */;
*  ANALYZE AFFECTS OF TIME SPENT ON ANALYSIS */;
*;
DATA ANALRT;
SET FP.WORKS2;
PLOTMRE = X_MRE_AC * 20;
PROC PLOT;
PLOT X_ANALRT * Y_ANALRT = PLOTMRE;
TITLE 'Effect of Analysis Rate on MRE (MRE X 20)';
*;
DATA RATE1;
SET FP.WORKS2;
IF X_ANALRT < 60 AND Y_ANALRT < 60;
PROC CORR SPEARMAN;
VAR X_AFPT Y_AFPT;
```

```
TITLE 'Analyses w/ Analysis Rates < 60';
PROC MEANS;
VAR X_MRE_AC Y_MRE_AC X_MRE4AC Y_MRE4AC;
*;
DATA RATE2;
SET FP.WORKS2;
IF X_ANALRT GE 60 AND Y_ANALRT GE 60 AND
   X_ANALRT < 150 AND Y_ANALRT < 150;
PROC CORR SPEARMAN;
VAR X_AFPT Y_AFPT;
TITLE 'Analyses w/ 60 <= Analysis Rates < 150';
PROC MEANS;
VAR X_MRE_AC Y_MRE_AC X_MRE4AC Y_MRE4AC;
*;
DATA RATE3;
SET FP.WORKS2;
IF X_ANALRT GE 150 AND Y_ANALRT GE 150;
PROC CORR SPEARMAN;
VAR X_AFPT Y_AFPT;
TITLE 'Analyses w/ Analysis Rates >= 150';
PROC MEANS;
VAR X_MRE_AC Y_MRE_AC X_MRE4AC Y_MRE4AC;
*;
DATA RATE4;
SET FP.WORKS2;
IF (X_ANALRT < 60 AND Y_ANALRT GE 60) OR
   (X_ANALRT GE 150 AND Y_ANALRT < 150) OR
   (X_ANALRT GE 60 AND X_ANALRT < 150 AND
    Y_ANALRT < 60 AND Y_ANALRT GE 150);
PROC CORR SPEARMAN;
VAR X_AFPT Y_AFPT;
TITLE 'Analyses w/ Mixed Analysis Rates I';
PROC MEANS;
VAR X_MRE_AC Y_MRE_AC X_MRE4AC Y_MRE4AC;
*;
DATA RATE5;
SET FP.WORKS2;
IF X_ANALRT < 70 AND Y_ANALRT < 70;
PROC CORR SPEARMAN;
VAR X_AFPT Y_AFPT;
TITLE 'Analyses w/ Analysis Rates < 70';
PROC MEANS;
VAR X_MRE_AC Y_MRE_AC X_MRE4AC Y_MRE4AC;
*;
DATA RATE6;
SET FP.WORKS2;
IF (X_ANALRT GE 70 AND Y_ANALRT GE 70);
PROC CORR SPEARMAN;
VAR X_AFPT Y_AFPT;
TITLE 'Analyses Analysis Rates >= 70';
PROC MEANS;
VAR X_MRE_AC Y_MRE_AC X_MRE4AC Y_MRE4AC;
*;
DATA RATE7;
SET FP.WORKS2;
IF (X_ANALRT < 70 AND Y_ANALRT GE 70) OR
   (X_ANALRT GE 70 AND Y_ANALRT < 70);
```

```
PROC CORR SPEARMAN;
VAR X_AFPT Y_AFPT;
TITLE 'Analyses w/ Mixed Analysis Rates II';
PROC MEANS;
VAR X_MRE_AC Y_MRE_AC X_MRE4AC Y_MRE4AC;

PROC DELETE DATA = ANALRT RATE1 RATE2 RATE3 RATE4 RATE5 RATE6 RATE7;
*;
data stophere;
abort return;
*  ============================================================  */;
*  ANALYZE AFFECTS OF SIZE OF APPLICATION */;
*;
DATA TEMP1/*  CREATE AVERAGE SIZE VARIABLE  */;
SET FP.WORKS2;
AVGSIZE = (X_AFPT + Y_AFPT) / 2;
*;
DATA SIZE1;
SET TEMP1;
IF AVGSIZE < 225;
PROC CORR SPEARMAN;
VAR X_AFPT Y_AFPT;
TITLE 'Analyses < 225 FPTS';
PROC MEANS;
VAR X_MRE_AC Y_MRE_AC X_MRE4AC Y_MRE4AC;
*;
DATA SIZE2;
SET TEMP1;
IF AVGSIZE GE 225 AND AVGSIZE < 475;
PROC CORR SPEARMAN;
VAR X_AFPT Y_AFPT;
TITLE 'Analyses >= 225 and < 475';
PROC MEANS;
VAR X_MRE_AC Y_MRE_AC X_MRE4AC Y_MRE4AC;
*;
DATA SIZE3;
SET TEMP1;
IF AVGSIZE GE 475;
PROC CORR SPEARMAN;
VAR X_AFPT Y_AFPT;
TITLE 'Analyses >= 475';
PROC MEANS;
VAR X_MRE_AC Y_MRE_AC X_MRE4AC Y_MRE4AC;
*;
DATA SIZE4;
SET TEMP1;
IF AVGSIZE < 320;
PROC CORR SPEARMAN;
VAR X_AFPT Y_AFPT;
TITLE 'Analyses < 320 FPTS';
PROC MEANS;
VAR X_MRE_AC Y_MRE_AC X_MRE4AC Y_MRE4AC;
*;
DATA SIZE5;
SET TEMP1;
IF AVGSIZE GE 320;
PROC CORR SPEARMAN;
```

```
VAR X_AFPT Y_AFPT;
TITLE 'Analyses >= 320';
PROC MEANS;
VAR X_MRE_AC Y_MRE_AC X_MRE4AC Y_MRE4AC;

PROC DELETE DATA = TEMP1 SIZE1 SIZE2 SIZE3 SIZE4 SIZE5;
*;
*  ==============================================================  */;
*;
*  ANALYZE AFFECT OF EXPERIENCE  */;
*;
DATA EXPER1;
SET FP.WORKS2;
IF (X_FPCTTM < 2 AND Y_FPCTTM < 2) AND
   (X_APPDTM < 2 AND Y_APPDTM < 2);
PROC CORR SPEARMAN;
VAR X_AFPT Y_AFPT;
TITLE 'Analyses by Neophyte Raters';
PROC MEANS;
VAR X_MRE_AC Y_MRE_AC X_MRE4AC Y_MRE4AC;
*;
DATA EXPER2;
SET FP.WORKS2;
IF X_FPCTTM GE 2 AND Y_FPCTTM GE 2;
PROC CORR SPEARMAN;
VAR X_AFPT Y_AFPT;
TITLE 'Analyses by Experienced Raters';
PROC MEANS;
VAR X_MRE_AC Y_MRE_AC X_MRE4AC Y_MRE4AC;
*;
DATA EXPER3;
SET FP.WORKS2;
IF (X_FPCTTM < 2 AND Y_FPCTTM < 2) AND
   (X_APPDTM GE 2 AND Y_APPDTM GE 2);
PROC CORR SPEARMAN;
VAR X_AFPT Y_AFPT;
TITLE 'Analyses by IS but not FP Experienced Raters';
PROC MEANS;
VAR X_MRE_AC Y_MRE_AC X_MRE4AC Y_MRE4AC;
*;
DATA EXPER4;
SET FP.WORKS2;
IF (X_FPCTTM < 2 AND Y_FPCTTM GE 2) OR
   (X_FPCTTM GE 2 AND Y_FPCTTM < 2);
PROC CORR SPEARMAN;
VAR X_AFPT Y_AFPT;
TITLE 'Analyses By Raters With Mixed Experience Levels';
PROC MEANS;
VAR X_MRE_AC Y_MRE_AC X_MRE4AC Y_MRE4AC;
*;
DATA TEMP;
SET FP.WORKS2;
PLOTMRE = X_MRE_AC * 20;
PROC PLOT;
PLOT X_FPCTTM * Y_FPCTTM = PLOTMRE;
TITLE 'Effect of Experience on MRE (MRE X 20)';
*;
```

```
PROC DELETE DATA = EXPER1 EXPER2 EXPER3 EXPER4 TEMP;
*;
*  ==============================================================  */;
*  ANALYZE AFFECT OF APPLICATION TYPE  */;
*;
DATA TEMP1;
SET FP.WORKS2;
IF X_APTYP2 = 1 AND Y_APTYP2 = 1;
PROC CORR SPEARMAN;
VAR X_AFPT Y_AFPT;
* TITLE 'Analyses of Acct'g/Finance Applics';
PROC MEANS;
VAR X_MRE_AC Y_MRE_AC X_MRE4AC Y_MRE4AC;
*;
DATA TEMP2;
SET FP.WORKS2;
IF X_APTYP2 > 1 AND Y_APTYP2 > 1;
PROC CORR SPEARMAN;
VAR X_AFPT Y_AFPT;
* TITLE 'Analyses of NON-Acct'g/Fin. Applics';
PROC MEANS;
VAR X_MRE_AC Y_MRE_AC X_MRE4AC Y_MRE4AC;
*;
DATA TEMP3;
SET FP.WORKS2;
IF (X_APTYP2 > 1 AND Y_APTYP2 = 1) OR
   (X_APTYP2 = 1 AND Y_APTYP2 > 1);
PROC CORR SPEARMAN;
VAR X_AFPT Y_AFPT;
* TITLE 'Analyses of Mixed Types of Applications';
PROC MEANS;
VAR X_MRE_AC Y_MRE_AC X_MRE4AC Y_MRE4AC;
*;
PROC DELETE DATA = TEMP1 TEMP2 TEMP3;
*;
*  ==============================================================  */;
*  COUNTING RATE V. EXPERIENCE  */;
*;
DATA COUNT;
MERGE FP.FPANAL3 FP.RATERBK2;
PROC PLOT;
PLOT ANALRATE * FPCT_TIM;
TITLE 'Experience (yrs) v. Counting Rate (FP/hr)';
*;
                              /*  THE END  */;
```