Reliability of Function Points Measurement

A FIELD EXPERIMENT

Chris F. Kemerer

Software engineering management encompasses two major functions, planning and control, both of which require the capability to accurately and reliably measure the software being delivered. Planning of software development projects emphasizes estimation of appropriate budgets and schedules. Control of software development requires a means to measure progress on the project and to perform after-the-fact evaluations of the project, for example, to evaluate the effectiveness of the tools and techniques employed on the project to improve productivity.

BOX A.

Function Points Calculation

Readers interested in learning how to calculate Function Points are referred to one of the fully documented methods, such as the IFPUG Standard, Release 3.0 [27]. The following is a minimal description only. Calculation of Function Points begins with counting five components of the proposed or implemented system, namely, the number of external inputs (*e.g.*, transaction types), external outputs (*e.g.*, report types), logical internal files (files as the user might conceive of them, not physical files), external interface files (files accessed by the application but not maintained, *i.e.*, updated by it), and external inquiries (types of on-line inquiries supported). Their complexity is classified as being relatively low, average, or high, according to a set of standards that define complexity in terms of objective guidelines. Table A.1 is an example of such a guideline, in this case the table used to assess the relative complexity of External Outputs, such as reports.

To use this table in counting the number of FPs in an application, a report would first be classified as an External Output. By determining the number of unique files used to generate the report ("File Type Referenced"), and the number of fields on the report ("Data Element Types"), it can be classified as a relatively low-, average-, or high-complexity External Output. After making such determinations for each of the five component types, the number of each component type present is placed into its assigned cell next to its weight in the matrix shown in Table A.2. Then, the total number of function counts (FCs) is computed as shown in Equation (1).

$$FC = \sum_{i=1}^{5} \sum_{j=1}^{3} w_{ij} x_{ij} \quad (1)$$

where w_{ij} = weight for row i, column j, and x_{ij} = value in cell i, j.

The second step involves assessing the impact of 14 general system characteristics that are rated on a scale from 0 to 5 in terms of their likely effect for the system being counted. These characteristics are: (1) data communications, (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, and (14) facilitates change. These values are summed and modified then to compute the Value Adjustment Factor, or VAF:

$$VAF = 0.65 + 0.01 \sum_{i=1}^{14} c_i \quad (2)$$

where c_i = value for general system characteristic i, for $0 <= c_i <= 5$.

Finally, the two values are multiplied to create the number of Function Points (FP):

$$FP = FC\ (VAF). \quad (3)$$

Table A.1. Complexity Assignment for External Outputs [27]

	1–5 Data Element Types	6–19 Data Element Types	20+ Data Element Types
0–1 File Types Referenced	Low	Low	Average
2–3 File Types Referenced	Low	Average	High
4+ File Types Referenced	Average	High	High

Table A.2. Function Count Weighting Factors

	Low	Average	High
External Input	__ × 3	__ × 4	__ × 6
External Output	__ × 4	__ × 5	__ × 7
Logical Internal File	__ × 7	__ × 10	__ × 15
External Interface File	__ × 5	__ × 7	__ × 10
External Inquiry	__ × 3	__ × 4	__ × 6

Unfortunately, as current practice often demonstrates, both of these activities are typically not well performed. Software projects often run from 100 to 200+% over budget, due both to inadequate initial estimates and to managers' inability to accurately monitor the project's progress, owing in part to a lack of objective measures [19, 24]. Accurate measures of the complexity-adjusted size of the deliverables of a software project early in the lifecycle will permit the estimation of the relationships between the deliverables and the cost and time required to produce them. However, any error in the measurement of the deliverables will add to the errors involved in estimating the required resources. Therefore, a critical first step in software management is the use of reliable software size measures.

Current Measures for Software Management

While a large academic literature exists on software measures/metrics, there are essentially only two software size measures that are widely used in practice for software planning and control. These are the number of source lines of code (SLOC) delivered in the final system and the number of Function Points. SLOC, the older of the two measures, has been criticized in both its planning and control applications. In planning, the task of estimating the final SLOC count for a proposed system has been shown to be difficult to do accurately in practice [18]. In control applications, SLOC measures for evaluating productivity have weaknesses as well, in particular, the problem of comparing systems written in different languages [16].

An alternative software size measure was developed by Allan Albrecht of IBM [1, 2]. This measure, which he termed "function points" (hereafter FPs), is designed to size a system in terms of its delivered functionality, measured in terms of such objects as the numbers of inputs, outputs, and files.[1] Albrecht argued that these entities would be much easier to estimate than SLOC early in the software project lifecycle and would be generally more meaningful to nonprogrammers. In addition, for evaluation purposes, they would avoid the difficulties involved in comparing SLOC counts for systems written in different languages.

FPs have proven to be a broadly popular measure with both practitioners and academic researchers. Dreger [14] estimates that some 500 major corporations worldwide are using FPs, and, in a survey by the Quality Assurance Institute, FPs were found to be regarded as the best available MIS productivity measure [20]. They have also been widely used by researchers in such applications as cost estimation [17], software development productivity evaluation [5, 25], software maintenance productivity evaluation [4], software quality evaluation [11], and software project sizing [3].

Research Questions in FP Reliability

Despite their wide use by researchers and their growing acceptance in practice, FPs are not without criticism. The first criticism revolves around the alleged low *interrater reliability* of FP counts, that is, whether two individuals performing an FP count for the same system would generate the same result. The author of a leading software engineering textbook summarizes his discussion of FPs as follows: "The function-point metric, like LOC, is relatively controversial . . . Opponents claim that the method requires some 'sleight of hand' in that computation is based on subjective, rather than objective, data . . ." [21, p. 94].

This perception of FPs as being unreliable has undoubtedly slowed their acceptance as a measure, as both practitioners and researchers may feel that in order to ensure sufficient measurement reliability either (a) a single individual would be required to count all systems or (b) multiple raters should be used for all systems and their counts averaged to approximate the "true" value [28]. Both of these options are unattractive in terms of either decreased flexibility or increased cost.

A second, related concern has developed more recently, due in part to FPs' growing popularity. A number of researchers and consultants have developed variations on the original method developed by Albrecht [13, 14, 23, 28] (also, C. Jones, Software Productivity Research, Inc., Feb. 20, 1988, mimeo, version 2). A possible concern with these variants is that counts using these methods may differ from counts using the original method [22, 30]. Jones has compiled a list consisting of 14 named variations and suggests that the values obtained using these variations might differ by as much as plus or minus 50% from the original Albrecht method (Software Productivity Research, Inc., Dec. 9, 1989, mimeo). If true, this lack of *intermethod reliability* poses several practical problems. From a planning perspective, one problem would be that for organizations adopting a method other than the Albrecht standard, the data they collect may not be consistent with those used in the development and calibration of a number of estimation models, (*e.g.*, see [2] and [17]). If the organization's data were not consistent with this previous work, then the estimated parameters of those models would no longer be directly usable by the organization. This would then force the collection of a large, internal dataset before FPs could be used to aid in cost and schedule estimation, which would involve considerable extra delay and expense. A second problem would be that for organizations that had previously adopted the Albrecht standard and desired to switch to another variation, the switch might render previously developed models and heuristics less accurate. From a control perspective, organizations using a variant method would have difficulty in comparing their *ex post* FP productivity rates to those of other organizations. For organizations that switched methods, the new data might be sufficiently inconsistent as to render trend analysis meaningless.

Finally, a related practical concern is the labor-intensive nature of FP counting. The originally developed procedure does not lend itself easily to automated data collection, and therefore another motivation for variant-counting methods is to develop an approach that would be automatable, perhaps through the

[1] Readers unfamiliar with FPs are referred to Box A for an overview of FP definitions and calculations.

use of computer-aided software engineering (CASE) tools. However, the question of the reliability of these new methods with the standard method remains. The conclusion of the preceding discussion is that the possibility of significant variations across methods poses a number of practical concerns, and there are currently only limited research results with which to guide practice in this area.

This article addresses the following specific research questions:

1. What is the interrater reliability of the standard FP-counting method?
2. What is the interrater reliability of a newer, alternative counting method?
3. What is the intermethod reliability of these two methods?

The approach taken was a field experiment involving more than 100 different total counts in a dataset with up to 27 actual commercial systems. Multiple raters and two methods were used to generate multiple counts of the systems, whose average size was 450 FPs. Briefly, the results of the study were (1) that the median difference in FP counts from pairs of raters using the standard method was approximately 12% and (2) that the correlation across the two methods was as high as 0.95 for the data in this sample. These results provide project managers with (1) objective measures of the degree of reliability of the measure and (2) evidence that the intermethod reliability is sufficiently high as to allow substitution of methods.

Research Design and Previous Research

Despite both the widespread use of FPs and the attendant criticism of their suspected lack of reliability, there has been only limited research on either the interrater question or the intermethod question. Perhaps the first attempt at investigating the interrater reliability question was made by members of the IBM GUIDE Productivity Project Group, the results of which are described by Rudolph as follows:

"In a pilot experiment conducted in February 1983 by members of the GUIDE Productivity Project Group . . . about 20 individuals judged independently the function point value of a system, using the requirement specifications. Values within the range +/− 30% of the average judgment were observed . . . The difference resulted largely from differing interpretation of the requirement specification. This should be the upper limit of the error range of the function point technique. Programs available in source code or with detailed design specification should have an error of less than +/− 10% in their function point assessment. With a detailed description of the system there is not much room for different interpretations" [25, p. 6].

Aside from this description, there has been no documented research until the study by Low and Jeffery [18], the first widely available, well-documented study of this question. Their research addressed only one of the two issues relevant to the current research, interrater reliability of FP counts. Their research methodology was a lab experiment using professional systems developers as subjects, with the unit of analysis being a set of program-level specifications. Two sets of program specifications were used in the experiment, both of which had been pretested with student subjects. For the interrater reliability question, 22 systems development professionals who counted FPs as part of their employment in 7 Australian organizations were used, as were an additional 20 inexperienced raters who were given training in the then-current Albrecht standard. Each of the experienced raters used his or her organization's own variation on the Albrecht standard (personal correspondence, R. Jeffery, Aug. 15, 1990). With respect to the interrater reliability research question Low and Jeffery found that the consistency of FP counts "appears to be within the 30 percent reported by Rudolph" within organizations, i.e., using the same method [18, p. 71].

Design of the Study

Given the Low and Jeffery research, a deliberate decision was made at the beginning of the current research to select an approach that would complement their work by (*a*) addressing the interrater reliability question using a different design and by (*b*) directly focusing on the intermethod reliability questions. The current work is designed to strengthen the understanding of the reliability of FP measurement, building on the base started by Low and Jeffery.

The area of overlap is the question of interrater reliability. Low and Jeffery chose a small group experiment, with each subject's identical task being to count the FPs implied from the two program specifications. Due to this design choice, the researchers were limited to choosing relatively small tasks, with the mean FP size of each program being 58 and 40 FPs, respectively. A possible concern with this design would be the external validity of the results obtained from the experiment in relation to real-world systems. Typical medium-sized application systems are generally an order of magnitude larger than the programs counted in the Low and Jeffery experiment [15, 29]. Readers whose intuition is that FPs are relatively unreliable might argue that the unknown true reliability is worse than that estimated in that experiment, since presumably it is easier to understand, and therefore count correctly, a small problem than a large one. On the other hand, readers whose intuition is that the unknown true reliability is better than that estimated in the experiment might argue that the experiment may have underestimated the true reliability since a single error, such as omitting one file type, would have a larger percentage impact on a small total than a large one. Finally, a third opinion might be that both effects are present but that they cancel each other out, and therefore the experimental estimates are likely to be representative of the reliability of counts of actual systems. Given these competing arguments, validation of the results on larger systems is clearly indicated. Therefore, one parameter for the research design was to test interrater reliability using actual average-sized application systems.

A second research design question suggested by the Low and Jeffery results, but not explicitly tested by

them, is the question of intermethod reliability. Reliability of FP counts was greater within organizations than across them, a result attributed by Low and Jeffery to possible variations in the methods used (personal correspondence, Aug. 15, 1990). As discussed earlier, Jones has also suggested the possibility of large differences across methods (Software Productivity Research, Inc., Dec. 9, 1989, mimeo). Given the growing proliferation of variant methods this question is also highly relevant to the overall question of FP reliability.

The goal of estimating actual medium-sized application systems required a large investment of effort on the part of the organizations and individuals participating in the research. Therefore, this constrained the test of intermethod reliability to a maximum of two methods to assure sufficient sample size to permit statistical analysis. The two methods chosen were (1) the International Function Point Users Groups (IFPUG) standard Release 3.0, which was the latest release of the original Albrecht method, [27] and (2) the Entity-Relationship approach developed by Desharnais [13].

The choice of the IFPUG 3.0-Albrecht Standard method (hereafter the "Standard method") was relatively obvious, as it is the single most widely adopted approach in current use, due in no small part to its adoption by the over 300-member IFPUG organization. Therefore, there is great practical interest in knowing the interrater reliability of this method. The choice of a second method was less clear-cut, as there are a number of competing variations. Choice of the Entity-Relationship method (hereafter "ER method") was suggested by a second concern often raised by practitioners. In addition to possible concerns about reliability, a second explanation for the reluctance to adopt FPs as a software measure is the perception that FPs are relatively expensive to collect, given the current reliance on labor-intensive methods [6]. Currently, there is no fully automated FP-counting system, in contrast to many such systems for the competing measure, SLOC. Therefore, many organizations have adopted SLOC not due to a belief in greater benefits, but due to the expectation of lower costs in collection. Given this concern, it would be highly desirable for there to be a fully automated FP collection system, and vendors are currently at work developing such systems. One of the necessary preconditions for such a system is that the design-level data necessary to count FPs be available in an automated format. One promising first step toward developing such a system is the notion of recasting the original FP definitions in terms of the widely used ER data-modeling approach. Many of the CASE tools that support data modeling explicitly support the ER approach, and therefore an FP method based on ER modeling seems to be a highly promising step toward the total automation of FP collection. Therefore, for all of the reasons stated above, the second method chosen was the ER approach.[2]

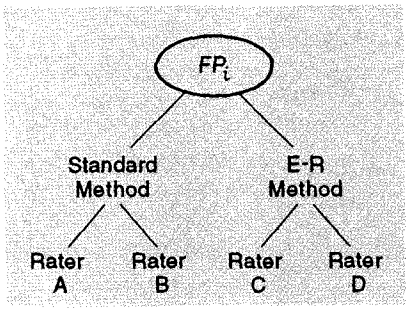


Figure 1. Overall research design

In order to accommodate the two main research questions, interrater reliability and intermethod reliability, the research design depicted in Figure 1 was developed and executed for each system in the dataset.

For each system i to be counted, four independent raters from that participating organization were assigned, two of them to the Standard method and two of them to the ER method. These raters were identified as Raters A and B (Standard method) and Raters C and D (ER method) as shown in Figure 1.

The definition of reliability used in this article is that of Carmines and Zeller, who define reliability as concerning "the extent to which an experiment, test, or any measuring procedure yields the same results on repeated trials . . . This tendency toward consistency found in repeated measurements of the same phenomenon is referred to as reliability" [7, pp. 11–12].

Allowing for standard assumptions about independent and unbiased error terms, the reliability of two parallel measures, x and x′, can be shown to be represented by the simple statistic, $\rho xx'$ [7]. Therefore, for the design depicted in Figure 1, the appropriate statistics are:[3]

$\rho(FP_{Ai}FP_{Bi})$ = interrater reliability for Standard method for System i
$\rho(FP_{Ci}FP_{Di})$ = interrater reliability for ER method for System i
$\rho(FP_{1i}FP_{2i})$ = intermethod reliability for Standard (1) and ER (2) methods for System i.

While this design addresses both major research questions, it is a very expensive design from a data collection perspective. Collection of FP counts for one medium-sized system was estimated to require four work hours on the part of each rater.[4] Therefore, the total data collection cost for each system, i, was estimated at 16 work hours, or two work days per system. A less expensive alternative would have been to use only two raters, each of whom would use one method and then recount using the second method, randomized for possible ordering effects. Unfortunately, this alternative design would suffer from a relativity bias, whereby raters would tend to remember the answer from their first count, and thus such a design would be likely to produce artificially high correlations [7, ch. 4]. Therefore, the more expensive design was chosen, with the foreknowledge that this would likely limit the number of organizations willing and able to participate, and therefore limit the sample size.

Data Collection

The pool of raters came from orga-

[2]Readers interested in the E-R approach are referred to [16]. However, a brief overview and example are provided in Box B.

[3]In order to make the subscripts more legible, the customary notation $\rho_{xx'}$ will be replaced with the parenthetical notation $\rho(xx')$.

[4]For future reference of other researchers wishing to replicate this analysis, actual reported effort averaged 4.45 hours per system.

nizations that are members of the International Function Point Users Group (IFPUG), although only a small fraction of the raters are active IFPUG members. The organizations represent a cross section of U.S., Canadian, and U.K. firms, both public and private, and represent a wide spectrum of industries. As per the research agreement, their actual identities will not be revealed; however, characterizations of participants by industry SIC codes and by system type are shown in Box C. The first step in the data collection procedure was to send a letter to an information systems contact person at each organization explaining the research and inviting participation. The contacts were told that each system would require four independent counts, at an estimated effort of four hours per count. Based on this mailing, information systems contacts at

Box B.

Entity-Relationship Approach Summary

The following material is excerpted directly from the materials used by Raters C and D in the experiment and highlights the general approach taken in the ER approach to FP counting. Readers interested in further details regarding the experimental materials should see [9], and for complete details regarding the ER approach see [13].

"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. . . . 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 subfunction 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)."

The following is an example of the documentation provided to count one of the five function types, Internal Logical Files.

"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:

Guidance.

- Entities updated by application are counted as logical internal files.
- Complexity is based on the number of relationships in which the entity participates as well as the number of DETs.
- 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."

Table B.1. Complexity Assignment for Internal Logic Files, ER Method

	1–19 Data Attribute Types in the Entity	20–50 Data Attribute Types in the Entity	51+ Data Attribute Types in the Entity
1 Relationship or other Entity Type	Low	Low	Average
2–5 Relationships or other Entity Types	Low	Average	High
6+ Relationships or other Entity Types	Average	High	High

63 organizations expressed interest in the research and were sent a packet of research materials. The contacts were told to select recently developed medium-sized applications, defined as those that required from one to six work years of effort to develop. After a follow-up letter, and, in some cases, follow-up telephone call(s), usable data were ultimately received on 27 systems. The only direct benefit promised to the participants was a report comparing their data with the overall averages. Using the classification scales developed by Jones [16] the vast majority of applications can be described as being interactive MIS-type systems, typically supporting either Accounting/Finance or Manufacturing-type applications (Box C).

Experimental Controls

A number of precautions were taken to protect against threats to validity, the most prominent being the need to ensure that the four counts were done independently. First, in the instructions to the site contact the need for independent counts was repeatedly stressed. Second, the packet of research materials contained four separate data collection forms, each uniquely labeled A, B, C, and D for immediate distribution to the four raters. Third, four FP manuals were included, two of the Standard method (labeled Method I) and two of the ER method (labeled Method II). While increasing the reproduction and mailing costs of the research, it was felt that this was an important step to reduce the possibility of inadvertent collusion through the sharing of manuals across raters, where the first rater might make marginal notes or otherwise give clues to a second reader as to the first rater's count. Fourth, and finally, four individual envelopes, prestamped and preaddressed to the researcher, were enclosed so that immediately on completion of the task the rater could place the data collection sheet into the envelope and mail it to the research team in order that no postcount collation by the site contact would be required. Again, this added some extra cost to the research, but was deemed to be an important additional safeguard.

Box C.

Data Background

Table C.1. Data by Industry

Industry	Percentage
Conglomerate	16%
Agriculture, Forestry & Fishing	5%
Mining	5%
Construction	0%
Manufacturing	26%
Transportation, Communication, Electric, Gas & Sanitary	11%
Wholesale & Retail Trade	5%
Finance, Insurance & Real Estate	16%
Services	5%
Government	11%

Table C.2. Data by System Type

System Type (source: [16])	Percentage
Batch MIS application	15%
Interactive MIS application	70%
Scientific or mathematical application	0%
Systems software or support application/utility	11%
Communications of telecommunications application	0%
Embedded or real-time application	0%
Other or DNA	4%

Copies of all of these research materials are available in [9] for other researchers to examine and use if desired to replicate the study.

One additional cost to the research of these precautions to assure independence was that the decentralized approach led to the result that not all four counts were received from all of the sites. Table 1 summarizes the number of sets of data for which analysis of at least one of the research questions was possible.

In Table 1 the first column shows the type of data. The row labeled $A \wedge B$ indicates that data from both the A and B rater were received. Since both of these raters used the Standard method, the interrater reliability for this method can be assessed using these data. The second row is similar, except that it applies to the ER method. The third row refers to systems for which all four counts were received and can be used as originally designed to measure intermethod reliability. This set will be referred to as the *Quadset* to indicate that all four counts were present. The fourth row refers to systems for which at least one A or B count exists and at least one C or D count exists. These data can also be used to test intermethod reliability and will be

Table 1. Summary of Primary Data Collected

Counts Received:	Systems	Observations	Research Question
$A \wedge B$	27	54	Standard method Interrater reliability
$C \wedge D$	21	42	ER method Interrater reliability
$A \wedge B \wedge C \wedge D$	17	68	Intermethod reliability ("Quadset")
$(A \vee B) \wedge (C \vee D)$	26	90	Intermethod reliability ("Fullset")

referred to as the *Fullset.* The Fullset naturally includes all of the systems in the Quadset.

These counts reflect the data after the removal of five systems' data that were deemed unusable for purposes of the study. Data for two systems were not used as only one count for each system (an A in one case and a D in the other) was received, and therefore no comparison of any kind could be made. Data for two other systems, one an average of 3,590 FPs, and the other of 2,294 FPs, approximately 9.1 and 5.3 standard deviations above the mean for the interrater sample respectively, were also excluded, on the grounds that they reflected large systems rather than the medium-size (one to six work years) systems requested. Finally, data for a fifth system for which independence of the raters was in doubt were also excluded.[5]

Posttest of Random Assignment Assumption

Given that the four raters were assigned to one of the two methods by the site contact, one possible concern might be that their assignment may have been biased in some way. For example, if raters A and B had greater FP-counting experience, on average, than raters C and D, then any comparison of methods would be simultaneously testing the methods hypothesis and a hidden experience hypothesis [18]. Given the number of field sites involved, assignment of raters could not be rigorously controlled *a priori,* other than through the instructions given to the site contact. This lack of direct control over random assignment is typical in field experimentation [10, p. 6]. Therefore, *ex post* tests of independent variables that could be postulated to have some effect were done, and the results of these tests are presented in Tables 2 and 3.

As shown in Table 2, the average overall experience of the raters, in terms of their systems development experience, their experience in counting FPs, and the percentage of raters who were involved with the development or maintenance of the system being counted, was relatively consistent across all four groups. The results of one-way ANOVA tests for both rater differences and method differences (where A and B represent the Standard method and C and D represent the ER method) did not support rejecting the null hypothesis of zero difference between the mean levels of experience. In addition, the Scheffe multiple-comparison procedure was run on the full raters-nested-within-methods model, with the same result that no statistically significant difference was detectable at even the $\alpha = 0.10$ level for any of the possible individual cases (*e.g.,* A vs. B, A vs. C, A vs. D, B vs. C . . .) [26]. Therefore, later tests of possible methods effects on FP count data will be assumed to have come from randomly assigned raters with respect to relevant experience.

In addition to experience levels, another factor that might be hypothesized to affect FP measurement reliability might be the system source materials with which the rater has to work. As suggested by Rudolph [25], three levels of such materials might be available: (1) requirements analysis phase documentation, (2) external design phase documentation (e.g., hard copy of screen designs, reports, file layouts, and so forth), and (3) the completed system, which could include access to the actual source code. Each of the raters contributing data to the study was asked which of these levels of source materials he or she had access to in order to develop the FP count. The majority of all raters used design documentation ("level II"). However, some had access only to level-I documentation, and some had access to the full completed system, as indicated in Table 3. In order to assure that this mixture of source materials level was unbiased with respect to the assigned raters and their respective methods, ANOVA analysis as per Table 2 was done, and the results of this analysis are shown in Table 3.

Similar to the results for experience levels, it appears that access to source materials was sufficiently similar for each rater group as to rule this out as a probable source of bias. Therefore, later tests of possible methods effects on FP count data will be assumed to have come from randomly assigned raters with respect to source material.

Main Research Results

For each of the three research questions three sets of data are presented: (a) the average counts from each approach, (b) a Pearson correlation coefficient, and (c) a paired *t*-test of the null hypothesis of zero difference between the results.

1. Standard method.

$$H_0: \overline{FP}_A - \overline{FP}_B = 0$$

Based on the research design described earlier, the average value for the A raters was 436 (standard deviation of 345), and for the B raters it was 464 (383), with $n = 27$. The results of a test of interrater reliability for the standard method yielded a Pearson correlation coefficient (ρ) = 0.80, ($p = 0.0001$), suggesting a strong correlation between FP counts of two raters using the standard method. The results of a paired *t*-test of the null hypothesis that the difference between the means is equal to 0

[5]It should be noted that the correlations of the counts for two of these three latter systems were extremely high, and their exclusion in the interests of conservatism has the effect of reducing the overall reliability measures for the dataset.

was only −0.61 (p = 0.55), indicating no support for rejecting the null hypothesis. The power of this test for revealing the presence of a large difference, assuming it was to exist, is approximately 90% [8, Table 2.3.6].[6] Therefore, based on these results, there is clearly no statistical support for assuming the counts are significantly different.

2. Entity-Relationship method.

$$H_0: \overline{FP}_C - \overline{FP}_D = 0$$

The same set of tests was run for the two sets of raters using the ER method, *mutatis mutandis*. For $n = 21$, values of $\overline{FP}_C$ and $\overline{FP}_D$ were 476 (381) and 411 (323) respectively. Note that these values are not directly comparable to the values for $\overline{FP}_A$ and $\overline{FP}_B$, as they come from slightly different samples. The reliability measure is $\rho(FP_{Ci}\ FP_{Di}) = 0.74$ (p = 0.0001), not quite as high as for the Standard method, but nearly as strong a correlation. The results of an equivalent *t*-test yielded a value of 1.15 (p = 0.26), again indicating less reliability than the Standard method, but still well below the level where the null hypothesis of no difference might be rejected. The power of this test is approximately 82%.

3a. Intermethod reliability results.
Quadset analysis ($n = 17$)

The test of intermethod reliability is a test of the null hypothesis:

$$H_0: \overline{FP}_1 - \overline{FP}_2 = 0$$

$$\text{where } \overline{FP}_1 = \sum_{i=1}^{n} \frac{FP_{Ai} + FP_{Bi}}{2}$$

$$\text{and } \overline{FP}_2 = \sum_{i=1}^{n} \frac{FP_{Ci} + FP_{Di}}{2}$$

At issue here is whether FP raters using two variant FP methods will produce highly similar (reliable) results, in this particular case the two methods being the Standard method and the ER method. In the interests of conservatism, the first set of analyses uses only the 17 systems for which all four counts, A, B, C, and D, were obtained. This is to guard against the event, however unlikely, that the partial response systems were somehow different. The values for $\overline{FP}_1$ and $\overline{FP}_2$ were 418 (322) and 413 (288), respectively, and yielded a $\rho(FP_{1i}\ FP_{2i}) = 0.95$ (p = 0.0001). The *t*-test of the null hypothesis of no difference resulted in a value of 0.18 (p = 0.86), providing no support for rejecting the hypothesis of equal means. These results clearly speak to a very high intermethod reliability. However, the conservative approach of only using the Quadset data yielded a smaller sample size, thus reducing the power of the statistical tests (e.g., the relative power of this *t*-test is 74%). To increase the power of the test in order to ensure that the preceding results were not simply the result of the smaller sample, the next step replicates the analysis using the Fullset data, those for which at least one count from the Rater A and B method and at least one count from

[6] All later power estimates are also from this source, *loc. cit.*

Table 2. Check of Rater and Method Assignment Randomness, Experience

Experience Type:	A Raters Mean or %	B Raters Mean or %	C Raters Mean or %	D Raters Mean or %	ANOVA F-test, by Rater	ANOVA F-test, by Method	Scheffe Test, α = 0.10
Systems Development	11.3 yrs.	9.7 yrs.	10.9 yrs	11.2 yrs	F = 0.21 (p = 0.89)	F = 0.08 (p = 0.77)	Negative, all cases
Function Points	1.3 yrs.	1.5 yrs.	1.7 yrs	1.7 yrs	F = 0.41 (p = 0.75)	F = 0.96 (p = 0.33)	Negative, all cases
This Application System	6%	19%	15%	13%	F = 0.76 (p = 0.52)	F = 0.08 (p = 0.78)	Negative, all cases

Table 3. Check of Rater and Method Assignment Randomness, Materials

Source Materials Type:	A Raters %	B Raters %	C Raters %	D Raters %	ANOVA F-test, by Rater	ANOVA F-test, by Method	Scheffe Test, α = 0.10
Requirements Analysis Documentation (level I)	11%	6%	14%	14%	F = 0.37 (p = 0.78)	F = 0.82 (p = 0.37)	Negative, all cases
Detailed Design documentation (level II)	68%	66%	64%	67%	F = 0.03 (p = 0.99)	F = 0.03 (p = 0.87)	Negative, all cases
Completed System (level III)	21%	28%	23%	19%	F = 0.22 (p = 0.88)	F = 0.23 (p = 0.63)	Negative, all cases

the Rater C and D method were available.

3b. Intermethod reliability results. Fullset analysis ($n = 26$)

The results from the Fullset analysis, while somewhat less strong than the very high values reported for the Quadset, still show high correlation, and since the Fullset test has greater power to detect differences, should they exist, greater confidence can be placed in the result of no difference. The values of $\overline{FP_1}$ and $\overline{FP_2}$ were 403 (303) and 363 (252), respectively, and yielded a $\rho(FP_{1i}\ FP_{2i} = 0.84$ ($p = 0.0001$). The *t*-test of the null hypothesis was 1.25 ($p = 0.22$), with a power of 89%. Thus, it is still appropriate not to reject the null hypothesis of no difference across these two methods, and, based on the Fullset analysis, not rejecting the null hypothesis can be done with increased confidence.

Managerial Results and Discussion

From the statistical results summarized in Table 4 it can be concluded that both the interrater and intermethod reliability of FPs are high. From the point of view of practicing managers some additional information that might be helpful in using FPs is the average magnitude of the differences across raters and methods. For example, one use of such data would be in performing sensitivity analysis on FP counts that are used for performing project estimates. Given a single FP count, what might be an appropriate range to use to adjust for possible differences that may have resulted from getting an FP count from a different analyst?

1. Interrater results. A plot of the Rater A versus Rater B counts is shown as Figure 2. It should be noted that the dashed line is the 45-degree line representing a perfect match (A = B) rather than a line that has been fitted to the data. One clear outlier is present, but its data have not been excluded from any of the data analysis. As a practical test the percentage differences for the standard method across raters for any single pair of raters is simply

$$\left|\frac{FP_A - FP_B}{FP_A}\right|.$$

The median value for all 27 pairs of raters using the standard method is 12.22%. Due to the presence of a single large outlier, the mean value is greater (26.53%). Similarly, the interrater result for the ER method can be computed by substituting FP_C for FP_A and FP_D for FP_B. The median value for the 21 pairs of raters using the ER method is 20.66%. Again, the mean value is higher (38.85%). A plot of Rater C vs. Rater D is shown as Figure 3.

The interrater error for the ER method was almost twice that of the Standard method. There are a number of possible explanations for this difference. The first, and easiest to check, is whether the slightly different samples used in the analysis of the two methods (the 27 systems used by the Standard method and the 21 systems used by the ER method) may have influenced the results. To check this possibility, both sets of analyses were rerun, using only the Quadset of 17 systems for which all four counts were available. This subanalysis generated a median percentage error of 11.51% for the Standard method and a median percentage error of 20.66% for the ER method, so it appears as if the difference cannot simply be attributed to a sampling difference.

More likely explanations stem from the fact that the ER approach, while perhaps the most common data-modeling approach in current use, is still unfamiliar enough to cause errors. Of the raters contributing data to the study, 23% of the C and D raters reported having no prior experience or training in ER modeling, and thus were relying solely on the experimental documentation provided. Thus, the comparison of the Standard and ER methods results shows the combined effects of both the methods themselves, and their supporting manuals. Therefore, the possibility of the test materials, rather than the method *per se*, being the cause of the increased variation, cannot be ruled out by the study.

An additional hypothesis has been suggested by Allan Albrecht. He notes that the ER approach is a user functional view of the system, a view that is typically captured in the requirements analysis documentation, but sometimes does not appear in the detailed design documentation. To the degree that this is true, and to the degree that counters in this study used the detailed design documentation to the exclusion of using the requirements analysis documents, this may have hindered use of the ER method (personal correspondence, A.J. Albrecht, Sept., 1990). A similar possibility is that the application system's documentation used may not have contained ER diagrams, thus creating an additional intermediate step in the counting process for those raters using the ER method in order to create these diagrams, or their equivalents, which could have contributed to a greater number of er-

Table 4. Summary of Reliability Statistics

	Interrater, Standard method	Interrater, ER method	Intermethod, Quadset	Intermethod, Fullset
n (Systems, Counts)	27, 54	21, 42	17, 68	26, 90
$\rho(xy)$; p	0.80 (0.0001)	0.74 (0.0001)	0.95 (0.0001)	0.84 (0.0001)
paired *t*-test; p	−0.61 (0.55)	1.15 (0.26)	0.18 (0.86)	1.25 (0.22)
$1 - \beta$ (power)	0.90	0.82	0.74	0.89

rors and hence a wider variance. Finally, given that the raters typically had significant experience with prior IFPUG standards, their better performance using the Standard method may be partly attributable to their possibly greater comfort with this approach than the newer, ER approach.

Ultimately, the interrater reliability results for the ER method are the least practically meaningful of the three major results, as hand counting using the ER approach should be seen as only an intermediate step toward their eventual automation.

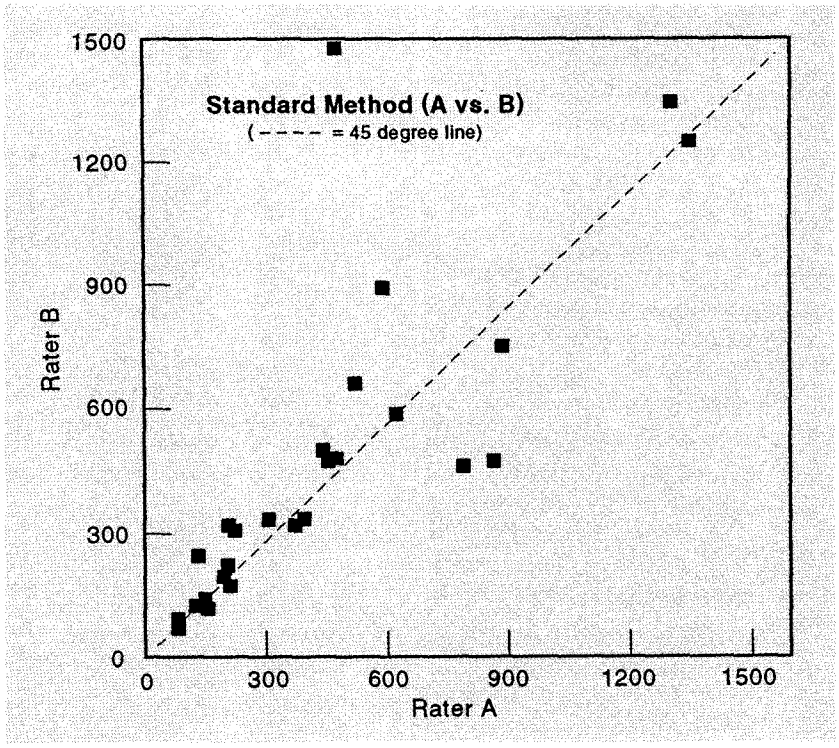
2. *Intermethod results.* The percentage error calculations for the intermethod results are, for the Quadset, a median of 17.95% (average = 17.75%) and for the Fullset, a median of 18.91% (average = 23.01%). Plots of the average Standard Method count vs. the average ER method count are shown as Figures 4 (Quadset) and 5 (Fullset). The intermethod results are the first documented study of this phenomenon and thus provide a baseline for future studies. The variation across the two methods is similar to that obtained across raters and thus does not appear to be a major source of error *for these two methods.* Of course, these results cannot necessarily be extended to pairwise comparisons of two other FP method variations, or even of one of the current methods and a third method. Determination of whether this result represents typical, better, or worse effects of counting variations must await further validation. However, as a practical matter, the results should be encouraging to researchers or vendors who might automate the ER method within a software engineering tool, thus addressing both the reliability and the data collection cost concerns. The results also suggest that organizations choosing to adopt the ER method now, although at some risk of possible lower interrater reliability, are likely to generate FP counts that are sufficiently similar to counts obtained with the Standard method so as to be a viable alternative. In particular, an analysis of the Quadset data revealed a mean FP count of 418 for the Standard method and 413 for the ER method, indistinguishable for both statistical and practical purposes.

Concluding Remarks

If software development is to fully establish itself as an engineering dis-

Figure 2. Interrater results, standard method (A vs. B)

Figure 3. Interrater results, E-R method (C vs. D)



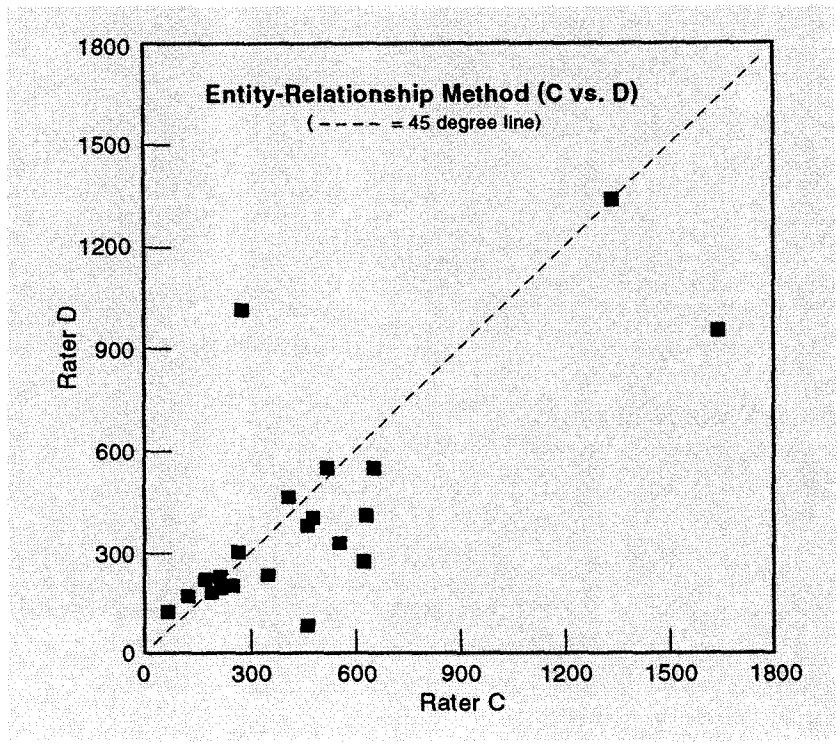
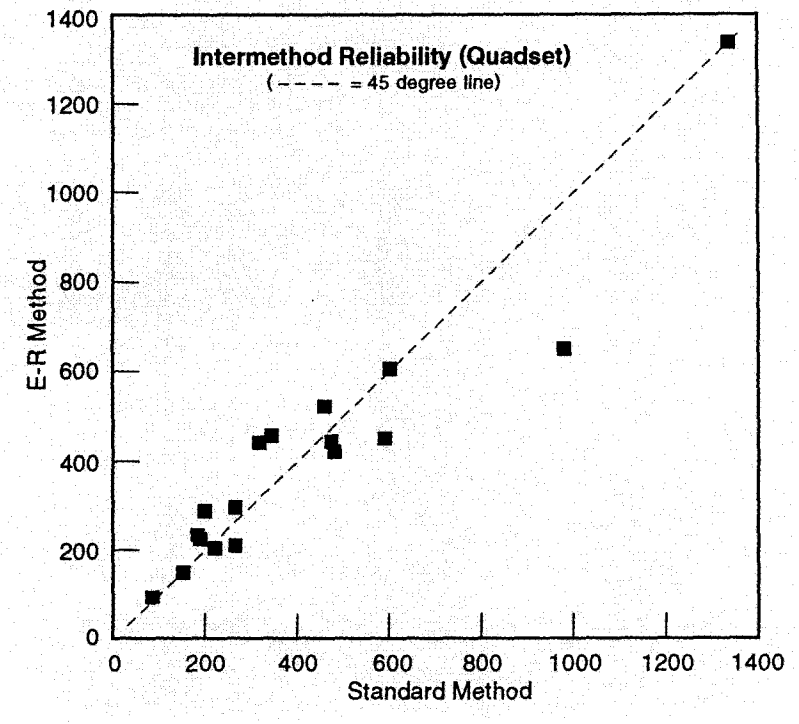
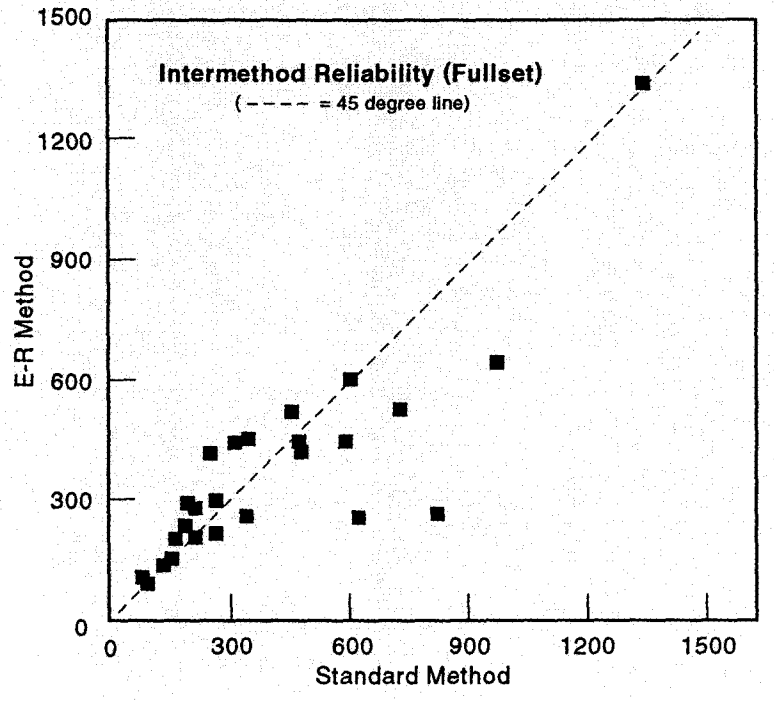


Figure 4. Intermethod results (Quadset)

Figure 5. Intermethod results (Fullset)





cipline, then it must adopt and adhere to the standards of such disciplines. A critical distinction between software engineering and other, more well-established branches of engineering, is the clear shortage of well-accepted measures of software. *Without* such measures the managerial tasks of planning and controlling software development and maintenance will remain stagnant in a 'craft'-type mode, whereby greater skill is acquired only through greater experience, and such experience cannot be easily communicated to the next project for study, adoption, and further improvement. *With* such measures software projects can be quantitatively described, and the managerial methods and tools used on the projects to improve productivity and quality can be evaluated. These evaluations will help the discipline grow and mature, as progress is made at adopting those innovations that work well, and discarding or revising those that do not.

Currently, the only widely available software measure that has the potential to fill this role for MIS projects in the near future is Function Points. This experiment has shown, contrary to some speculation and limited prior research, that both the interrater and intermethod reliability of FP measurement are sufficiently high that their reliability should not pose a practical barrier to their continued adoption and future development.

The collection effort for FP data in this research averaged approximately 1 work hour per 100 FPs and can be expected to be indicative of the costs to collect data in actual practice, since the data used in this research were actual commercial systems. For large systems this amount of effort is nontrivial and may at least partially account for the relative paucity of prior research on these questions. Clearly, further efforts directed toward developing aids to greater automation of FP data collection should continue to be pursued. However, even the current cost is small relative to the large sums spent on software development and maintenance in total, and managers should consider the time spent on FP collection and analysis as an investment in process improvement of their software development capability. Such investments are also indicative of true engineering disciplines, and there is increasing evidence of these types of investments in leading-edge software firms in the U.S. and in Japan [12]. Managers wishing to

quantitatively improve their software development and maintenance capabilities should adopt or extend software measurement capabilities within their organizations. Based on this experiment, FPs offer a reliable yardstick with which to implement this capability.

Acknowledgments

Helpful comments were received from A. Albrecht, N. Campbell, J. Cooprider, B. Dreger, P. Guinan, J. Henderson, R. Jeffery, C. Jones, M. Keller, W. Orlikowski, D. Reifer, A. Rollo, H. Rubin, E. Rudolph, W. Rumpf, G. Sosa, C. Symons, N. Venkatraman, and J. Verner. Finally, special thanks are due to my research assistant, M. Connolley. ◻

References

1. Albrecht, A.J. Measuring application development productivity. In *GUIDE/SHARE: Proceedings of the IBM Applications Development Symposium* (Monterey, Calif.), 1979, pp. 83–92.
2. Albrecht, A.J. and Gaffney, J. Software function, source lines of code, and development effort prediction: A software science validation. *IEEE Trans. Softw. Eng. SE-9,* 6 (1983), 639–648.
3. Banker, R.D. and Kemerer, C.F. Scale economies in new software development. *IEEE Trans. Softw. Eng. SE-15,* 10 (1989), 416–429.
4. Banker, R.D., Datar, S.M. and Kemerer, C.F. A model to evaluate variables impacting productivity on software maintenance projects. *Manage. Sci. 37,* 1 (1991), 1–18.
5. Behrens, C.A. Measuring the productivity of computer systems development activities with Function Points. *IEEE Trans. Softw. Eng. SE-9,* 6 (1983), 648–652.
6. Bock, D.B. and Klepper, R. FP-S: A simplified Function Point counting method. Working Paper, Southern Illinois Univ., at Edwardsville, Ill., 1990.
7. Carmines, E.G. and Zeller, R.A. *Reliability and Validity Assessment.* Sage Publications, Beverly Hills, Calif., 1979.
8. Cohen, J. *Statistical Power Analysis for the Behavioral Sciences.* Academic Press, New York, N.Y., 1977.
9. Connolley, M.J. An empirical study of Function Points analysis reliability. Masters thesis, MIT Sloan School of Management, Cambridge, Mass., 1990.
10. Cook, T.D. and Campbell, D.T. *Quasi-Experimentation: Design and Analysis Issues for Field Settings.* Houghton-Mifflin, Boston, 1979.
11. Cooprider, J. and Henderson, J. A multi-dimensional approach to performance evaluation for I/S development. Working Paper 197, MIT Center for Information Systems Research, Cambridge, Mass., 1989.
12. Cusumano, M. and Kemerer, C.F. A quantitative analysis of US and Japanese practice and performance in software development. *Manage. Sci. 36,* 11 (1990), 1384–1406.
13. Desharnais, J.-M. Analyse statistique de la productivite des projets de developpement en informatique a partir de la technique des points de fonction (English version). Masters thesis, Universite du Quebec, Montreal, 1988.
14. Dreger, J.B. *Function Point Analysis,* Prentice-Hall, Englewood Cliffs, N.J., 1989.
15. Emrick, R.D. Software development productivity second industry study. In the *1988 International Function Point Users Group Spring Conference Proceedings* (Dallas, Tex.). IFPUG, Westerville, Ohio, pp. 1–44.
16. Jones, C. *Programming Productivity.* McGraw-Hill, New York, 1986.
17. Kemerer, C.F. An empirical validation of software cost estimation models. *Commun. ACM 30,* 5 (May 1987), 416–429.
18. Low, G.C. and Jeffery, D.R. Function Points in the estimation and evaluation of the software process. *IEEE Trans. Softw. Eng. 16,* 1 (1990), 64–71.
19. Maglitta, J. It's reality time. *Computerworld* (1991), 81–84.
20. Perry, W.E. The best measures for measuring data processing quality and productivity. Tech. Rep. Quality Assurance Institute, 1986.
21. Pressman, R.S. *Software Engineering: A Practitioner's Approach.* McGraw-Hill, New York, 1987.
22. Ratcliff, B. and Rollo, A.L. Adapting Function Point analysis to Jackson system development. *Softw. Eng. J.* (1990), 79–84.
23. Rubin, H.A. Macroestimation of software development parameters: The estimacs system. In *IEEE SOFTFAIR Conference on Software Development Tools, Techniques and Alternatives.* IEEE, New York, N.Y., 1983.
24. Rubin, H.A. Measure for measure. *Computerworld* (1991), 77–79.
25. Rudolph, E.E. Productivity in computer application development. Working Paper 9, Univ. of Auckland, Dept. of Management Studies, Auckland, New Zealand, 1983.
26. Scheffe, H. *Analysis of Variance.* John Wiley & Sons, New York, 1959.
27. Sprouls, J. *IFPUG Function Point Counting Practices Manual Release 3.0.* International Function Point Users Group, Westerville, Ohio, 1990.
28. Symons, C.R. Function Point analysis: Difficulties and improvements. *IEEE Trans. Softw. Eng. 14,* 1 (1988), 2–11.
29. Topper, A. CASE: A peek at commercial developers uncovers some clues to the mystery. *Computerworld, 24,* 15 (1990), 61–64.
30. Verner, J.M., Tate, G., Jackson, B. and Hayward, R.G. Technology dependence in Function Point analysis: A case study and critical review. In *Proceedings of the 11th International Conference on Software Engineering.* 1989, pp. 375–382.

CR Categories and Subject Descriptors: D.2.8 [**Software Engineering**]: Metrics; D.2.9 [**Software Engineering**]: Management; K.6.0 [**Management of Computing and Information Systems**]: General—*Economics*; K.6.1 [**Management of Computing and Information Systems**]: Project and People Management; K.6.3 [**Management of Computing and Information Systems**]: Software Management

General Terms: Management, Measurement, Performance, Reliability

Additional Key Words and Phrases: Cost estimation, Entity-Relationship models, Function Points, productivity evaluation, project planning.

About the Author:

CHRIS F. KEMERER is the Douglas Drane Career Development Associate Professor of Information Technology and Management at the MIT Sloan School of Management. His research interests are in the measurement and modeling of software development. **Author's Present Address:** MIT Sloan School of Management, E53-315, 50 Memorial Drive, Cambridge, MA 02139.

Research support from the International Function Point Users Group and the MIT Center for Information Systems Research is gratefully acknowledged. Provision of the data was made possible in large part due to the efforts of A. Belden, B. Porter, and the organizations that contributed data to the study.