Improving the Reliability of Function Point Measurement: An Empirical Study

Chris F. Kemerer and Benjamin S. Porter

Abstract—**Information Systems development has operated for virtually its entire history without the quantitative measurement capability of other business functional areas such as marketing or manufacturing. Today, managers of information systems organizations are increasingly taken to task to measure and report, in quantitative terms, the effectiveness and efficiency of their internal operations. In addition, measurement of information systems development products is also an issue of increasing importance due to the growing costs associated with information systems development and maintenance.**

One measure of the size and complexity of information systems that is growing in acceptance and adoption is Function Points, a user-oriented, nonsource line of code metric of the systems development product. Recent previous research has documented the degree of reliability of Function Points as a metric. This research extends that work by a) identifying the major sources of variation through a survey of current practice, and b) estimating the magnitude of the effect of these sources of variation using detailed case study data from actual commercial systems. The results of this research show that a relatively small number of factors has the greatest potential for affecting reliability, and recommendations are made for using these results to improve the reliability of Function Point counting in organizations.

Index Terms—**Estimation, function points, management, measurement, performance, productivity evaluation, project planning, project control, reliability.**

I. Introduction

MANAGEMENT of software development and maintenance 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 the size of the system to be delivered in order that appropriate budgets and schedules can be agreed upon. Without valid size estimates, this process is likely to be highly inaccurate, leading to software that is delivered late and over-budget. Control of software development requires a means to measure progress on the project. In addition, measures are required to perform after-the-fact evaluations of the project such as evaluations of the tools and techniques employed on the project to improve productivity and quality.

Unfortunately, as current practice often demonstrates, both of these activities are typically not well performed, in part because of the lack of well-accepted measures, or metrics. Software size is a critical component of productivity and quality ratios, and has traditionally been measured by the number of source lines of code (SLOC) delivered in the final system. This metric 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 in practice to be difficult to do accurately [18]. In control, SLOC measures for evaluating productivity have weaknesses as well, in particular, the problem of comparing systems written in different languages [14].

Against this background, an alternative software size metric was developed by Allan Albrecht of IBM [2]. This metric, which he termed "function points" (FP's), was designed to size a system in terms of its delivered systems components, measured as a weighted sum of the number of inputs, outputs, inquiries, and files.[1] Albrecht argued that these components would be much easier to estimate than SLOC early in the software project life-cycle, and would generally be 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.

FP's have proven to be a widely accepted metric with both practitioners and academic researchers. Dreger estimates that some 500 major corporations world-wide are using FP's [11], and, in a survey by the Quality Assurance Institute, FP's were found to be regarded as the best available MIS productivity metric [19]. They have also been widely used by researchers in such applications as cost estimation [16], software development productivity evaluation [5], [22], software maintenance productivity evaluation [3], software quality evaluation [9], and software project sizing [4]. Additional work in defining standards has been done by Zwanzig [29] and Desharnais [10]. Although originally developed by Albrecht for traditional MIS applications, there has recently been significant work in extending FP's to scientific and real time systems [15], [21], [28].

Despite their wide use by researchers and their growing acceptance in practice, FP's are not without criticism. The main criticism revolves around the alleged low inter-rater reliability of FP counts, that is, whether two individuals performing a FP count for the same system would generate the same result [7]. The author of a leading software engineering textbook summarizes his discussion of FP's as follows.

Manuscript received October 1, 1991; revised August 1, 1992. This work was supported by the International Function Point Users Group and MIT's Center for Information Systems Research. Recommended by R. Selby and K. Torii.

C.F. Kemerer is with MIT, Cambridge, MA 02139.

B.S. Porter is with Andersen Consulting, Boston, MA 02110.

IEEE Log Number 9203768.

[1] Readers unfamiliar with FP's are referred to Appendix A for an overview of FP definitions and calculations.

0162–8828/92$03.00

"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 . . . " [20, p. 94].

This perception of FP's as being unreliable has undoubtedly slowed their acceptance as a metric, since 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 [26]. Both of these options are unattractive in terms of either decreased flexibility in the first case and likely increased cost and cycle times in the second.

This paper reports on the results of a two-phased research approach designed to investigate possible sources of unreliability in the FP measure. The first phase used a combination of key informants and a field survey to identify the most likely sources of FP counting variance. The second phase collected data from three detailed case studies which were then used to estimate the magnitude of the effect of the variations. In all, 33 FP counts were estimated from the detailed case study data.

The results from this analysis identified three potential sources of variation in FP counting: the treatment of backup files, menus, and external files used as transactions. These are the three areas where tighter standards are necessary and where managers should focus their attention on adopting and adhering to standard counting practices. The results of this research also identified several areas that have been suggested to cause variation, but may not be important sources of error in actual practice.

This paper is organized as follows. Section II presents a brief description of the research problem and the previous research. Section III describes the research methodology, which consisted of a survey and a set of quantitative case studies. Results of this analysis are presented in Section IV; Section V offers some concluding remarks.

II. Research Problem

2.1. Previous Research

Despite both the widespread use of FP's and some attendant criticism of their suspected lack of reliability, there has been little research on the reliability of FP's. Perhaps the first attempt at investigating this 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 Feb. 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" [22, p. 6].

Aside from this description, the first documented study was by Low and Jeffery [18]. Their research focused on the inter-rater reliability of FP counts using as their research methodology an experiment employing professional systems developers as subjects, with the unit of analysis being a set of program level specifications. Two sets of program specifications were used, both pre-tested with student subjects. For the inter-rater reliability question, 22 systems development professionals who counted FP's as part of their employment in seven 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 organization's own variation on the Albrecht standard [13]. With respect to the inter-rater reliability research question, Low and Jeffery found that the consistency of FP counts "appears to be within the 30% reported by Rudolph" within organizations [18, p. 71].

Most recently, Kemerer conducted a large-scale field experiment to address, among other objectives, the question of inter-rater reliability using a different research design [17]. Low and Jeffery chose a small group experiment, with each subject's identical task being to count the FP's implied from the two program specifications. Due to this design choice, they were limited to choosing relatively small tasks, with the mean FP size of each program being 58 and 40 FP's, respectively. Typical medium-sized application systems are generally an order of magnitude larger than the programs counted in that experiment [12], [27]. The Kemerer study tested inter-rater reliability using more than 100 different total counts in a data set with 27 actual commercial systems. Multiple raters were used to count the systems, whose average size was 450 FP's. The results of this analysis were that the pairs of FP counts were highly correlated ($\rho = 0.8$) and had a median variance of approximately 12%. These results suggest that FP's are much more reliable than previously suspected, and therefore may indicate that wider acceptance and greater adoption of FP's as a software metric is appropriate.

2.2. Current Research Problem

While both the Low and Jeffrey and the Kemerer results encourage the continued use of FP's, as the reliability is higher than previously speculated, there is clearly still room for improvement. In particular, given that one use of FP's is for managerial control of post-implementation productivity and quality evaluations, a 12% variance in counting could mask small but real underlying productivity changes, and therefore could interfere with proper managerial decision making. For example, a software project might have been a pilot test for use of a new tool or method, which resulted in a 10% productivity gain. If, through an unfortunate coincidence the output of this project was understated by 12%, then managers might come to the mistaken conclusion that the new tool or method had no, or even a slightly negative impact, and thus abandon it inappropriately.

Given this and similar scenarios, it is clearly important for management to have reliable instruments with which to measure their output. And, given that 1) FP's are already widely in use as a metric, and 2) have been shown to have good but imperfect reliability, it seems appropriate to attempt to determine the sources of the variation in counting as a first step toward eliminating those variances and making FP's an even more reliable metric.

III. Research Methodology

3.1. Introduction

The previous research described above used a large scale experimental design to identify the magnitude of the variations in FP counting. However, that research approach is ill-suited to the detailed analysis necessary to address the source of the variations in reliability. The current research consisted of two phases designed to identify the sources and magnitudes of decreased reliability of FP variations. In the first phase, key informants identified sixteen likely sources of variation. Then, a survey of 45 experienced users identified nine of these 16 as especially problematic. In the second phase, detailed quantitative case study data on three commercial systems were collected and each system was counted using each rule variation. These cases are from three distinct management information systems applications from three different organizations.

3.2. Survey Phase

Development of the survey form was accomplished with significant involvement of the Counting Practices Committee (CPC) of the International Function Points Users Group (IFPUG). The committee consists of approximately a dozen experts drawn from within the membership of IFPUG. IFPUG consists of approximately 350 member organizations worldwide, with the vast majority being from the United States and Canada [23]. IFPUG is generally viewed as the lead organization involved with FP measurement, and the CPC is the standards setting body within IFPUG [1].

The CPC is responsible for the publication of the Counting Practices Manual (hereafter referred to as CPM 3.0, as it was release 3.0 that was used in this research) [24]. This is IFPUG's definitive standards manual for the counting of FP's. In soliciting input from the CPC for this research, attention was focused on those systems areas for which a) no current standard exists in the CPM 3.0, and b) areas for which a standard exists, but for which there is believed to be significant noncompliance.

From a series of meetings and correspondence with these key informants an original survey of 14 questions was developed.[2] This survey was pre-tested with members of the CPC and a small number of IFPUG member organizations not represented on the CPC, which resulted in the addition of two questions and some minor changes to existing questions. The final 16-question survey is presented in Fig. 2. This survey was mailed to 84 volunteer member organizations of IFPUG, who were asked to document how FP counting was actually done within their organization. No compensation was provided for completing the survey, although respondents were promised a summary of the results. Completion of the survey was estimated to require one hour of an experienced FP counter's time. Forty-five usable surveys were received, for a response rate of 54%. No data are available on any possible nonresponse bias. In addition, it should be emphasized that this was not designed to be a random survey, but rather a survey of the population of IFPUG members, who are believed to represent, on average, organizations with greater knowledge and/or interest in FP counting as compared to organizations that have not joined the user group.

3.3. Case Study Phase

3.3.1. Introduction: While the survey phase of the research identified those areas that are likely sources of variation, it did not identify the magnitude of those effects. For example, while organizations may differ on the proper interpretation of a given FP construct, it may be the case that the situation described is relatively rare within actual information systems, such that differences in how it is treated may have negligible effect on an average FP count. Detailed data for each variant are required to assess the magnitude of the potential differences caused by each of the possible sources of variation. Given these data requirements, a quantitative case study methodology was chosen. As described by Swanson and Beath, this approach features the collection of multiple types of data, including documentation, archival records, and interviews [25]. The demand for detailed data with which to evaluate the multiple variations suggested by the surveys had two effects upon the research. First, a significant data collection and analysis effort was required for each case, since each variant required the collection of additional data and the development of a new FP count. Second, the detailed data requirements excluded a number of initially contacted organizations from participating in the final research.

The project selection criteria were that the projects were recently delivered and had a completed FP count in the range of 200–600 FP's.[3] This range was selected as encompassing medium sized application development and is the size range of the bulk of projects which are undertaken in North American systems development organizations today [11], [17]. Obtaining the final usable three sets of case study data required the solicitation of ten organizations. Only three of these possessed and were willing to share the necessary data with the researchers. These cases represent typical MIS applications systems that are the type for which FP's were developed, and which are

[2] It is interesting to note that all of these questions dealt with how to measure the five function count types, and none with the 14 complexity factors. This reflects the fact that any reliability concerns relating to the 14 complexity factors are small, given that their potential impact on the final FP count is constrained by the mathematical formula [2], [6]. Empirical research has also documented the result that the impact of the 14 complexity factors is small [16]. This is in contrast to the five function types, where the impact of a different interpretation is unconstrained, and can be potentially very large.

[3] For readers more familiar with SLOC size measures than with FP's, some comparative sizes may be useful. Jones reports an average conversion ratio of 106 SLOC of COBOL per FP and 320 SLOC of Assembly language per FP [14, p. 77].

TABLE I
PHASE I: SURVEY RESULTS

Question Number and Subject	Percent by Response category						
	1	2	3	4	5	Other	Candidate?
1. Backup Files	2%	4%	**40%**	16%	27%	11%	Yes
2. Multi-function External Output Screens	29%	7%	**42%**			22%	Yes
3.Error Messages	14%	32%	**21%**			34%	Yes
4. Menu Function Types	**37%**	7%	2%	2%	51%	0%	Yes
5. Menu Function Count	**37%**	16%	5%	40%		2%	Yes
6. Help Messages Function Count	9%	7%	**60%**	11%		13%	No
6a. Help Messages Function Type	0%	30%	**64%**			7%	No
7. Help Screen Function Count	4%	7%	**49%**	31%		8%	Yes
7a. Help Screen Function Type	2%	0%	2%	23%	**67%**	5%	No
8. Report with Detail and Subtotals	**89%**	5%	5%			2%	No
9. Hard Coded Tables	30%	7%	52%			11%	Yes
10. Report with 2 selection criteria	**59%**	36%				5%	No
11. Report ordered with 2 criteria	44%	**51%**				4%	No
12. External Inquiry Function count weights	**93%**	2%				5%	No
13. Logical Internal File used as transactions for another system	38%	9%	33%			20%	Yes
14. External Interface File used as External Inputs for a system	36%	13%	11%	29%		11%	Yes

representative of the type of systems developed and maintained by the original survey respondents. Each is described in more detail in the next section.

The results were obtained using a variance analysis approach. Each of the systems submitted for the analysis had the sites FP count and other relevant documentation. The analysis then systematically applied single variations of the counting rules which were identified in the research. These variations were those identified in the first phase for further analysis because they were different from the CPM 3.0 standard (or for which no standard had been established in the area), and they were being used by a significant population of the survey respondents.

IV. RESULTS

4.1. Survey Results

Table I contains the response data for the survey instrument in Fig. 2. The table summarizes the percentage of survey respondents selecting each of the possible answers. In each row the response that is in compliance with the CPM 3.0 is highlighted in bold. For example, for question number 1, 40% of the respondents chose answer 3, and noted that they "counted backup files as Logical Internal Files, but only when backup files were requested by the user or the auditors."

Given the extensive data collection and analysis requirements necessary to analyze each variant, the second phase (case study) of the research was designed to investigate only those topics identified by the first phase (survey) as the most likely sources of variance. In order to determine which topics merited further attention in the case studies, a target minimum was set equal to a 50% compliant response rate, i.e., the topics selected as candidates for further study were those where more than 50% of the responses were different from the CPM 3.0 standard. This cutoff, while arbitrary, was deemed appropriate given that these issues had been pre-selected as especially contentious.[4]

Therefore, the final column in Table I identifies those questions that provide the source for a candidate variant to be investigated in the second phase of the research. If the CPM 3.0 standard answer (the number in bold type), is less than 50%, then the topic was regarded as a candidate for further study. In addition, question numbers 9, 13, and 14 were also selected for further study. For question number 9 the CPM 3.0 does not contain a counting standard for this issue, and thus no CPM 3.0 compliant response is possible. For questions 13 and 14, the CPM 3.0 does have a standard. Unfortunately, upon analysis of the survey data it was determined that the survey questions were sufficiently ambiguous as to not clearly differentiate a single correct answer. Therefore, no CPM 3.0 "target" is shown for these two questions.

An additional interpretation of the data is that those questions for which the target answer is not the maximum answer

[4]From Table I it can be seen that the responses to two questions were near the cutoff point: number 7 with an agreement level of 49% and number 11 with an agreement level of 51%. To avoid *ex post* decision making with regard to the topics meriting further study, the original 50% guideline was strictly adhered to, with the result that question 7 was further investigated while question 11 was not.

in the row are those topics that IFPUG needs to better communicate the standard to FP counters. For these data those questions are numbers 3, 4 and 5.

4.1.2. Questions not Requiring Further Analysis: While discussions with the key informants of the standards setting committee suggested the 16 survey questions as potential areas for variance, the results of the survey showed that, for some questions, a majority of respondents were in compliance with the standards. Therefore, the results from these questions are only discussed here briefly, and were not the subject of the second phase of the research.

Responses to questions 8 and 12 were unique in their overwhelming adherence to the CPM 3.0. These questions were initially suggested by a definition of counting practices documented in a recent textbook [11]. The results of the survey indicate that these variations in counting standards are not widely used by the FP counters surveyed.

There were acceptable levels of agreement (>50%, as predetermined) among the respondents concerning questions 10 and 11, dealing with counting reports with multiple selection criteria and multiple sort sequences. The results of the survey were in compliance with the CPM 3.0 guidance as well. Therefore, no case studies were developed for these variations. Similarly, responses to questions 6 and 6a were also substantially in support of the CPM 3.0 standard. These related to the counting of "Help Messages" which may appear on various screens.

Responses to questions 7 and 7a also related to "Help Functions" but at the "Help Screen" level. There was less conformity as reflected by the response to question 7 (49% compliance with the CPM 3.0), but the response to 7a showed strong agreement with standards. Therefore, question 7 was deemed to merit further study, but question 7a was not.

4.1.3. Questions That are Candidates for Further Analysis: In the remaining nine questions (two with two possibilities each, for a total of eleven variants), there was significant noncompliance with the CPM 3.0 standards to warrant the further investigation of resulting potential variance from differing counting rule interpretations. These cases were identified by selecting the situations in which a majority of the respondents identified the use of a counting rule which was different from the CPM 3.0 standard, or for which no CPM 3.0 standard exists. Table II maps the 11 variants to the original survey questions.

Description of the 11 Variants: Variant 1: Counting Backup Files as Logical Internal Files—The CPM 3.0 requires counting these files as Logical Internal Files, but only if they are specifically requested by the user due to legal or other business requirements. As Logical Internal Files have the highest weighting factors in function point counting, counting the backup file as a Logical Internal File could have a significant impact on the overall FP count.

Variant 2: Counting Backup Files as External Outputs—About 20% of the respondents to the survey indicated that they count backup files as External Outputs. The weighting factors for External Outputs are less than Logical Internal Files, but could have a significant impact on overall FP counts if there were a large number of such files.

TABLE II
CASE STUDY TO SURVEY QUESTION MAPPING

Case Study Variants	Related Survey Questions
1. Counting Backup Files as Logical Internal Files	1. Backup Files
2. Counting Backup Files as External Outputs	1. Backup Files
3. Counting Add, Change, and Delete Outputs as separate functions	2.Multi-function External Output Screens
4. Counting Error Message Responses as individual data elements	3. Error Messages
5. Counting Menus as an External Inquiry	4. Menu Function Types
6. Counting Menus as one External Inquiry for each layer of menu	5. Menu Function Count
7. Counting Menus as one External Inquiry for each menu screen	5. Menu Function Count
8. Counting Help Screens as individual function types	7. Help Screen Function Count
9. Counting "Hard Coded" data tables as Logical Internal Files	9. Hard Coded Tables
10. Logical Internal File used as transactions for another system	13.Logical Internal File used as transactions for another system
11. Counting External Interface Files as External Inputs when used as transactions	14. External Interface File used as External Input transactions for a system

Variant 3: Counting Add, Change, and Delete Outputs as Separate Functions—CPM 3.0 counting rules allow the counting of each of the Add, Change, and Delete transactions as a separate function type. However, only 42% of the respondents indicated compliance. Organizations that do not count these separately may lose up to two-thirds of the points from External Inputs, and somewhat less from External Outputs.

Variant 4: Counting Error Message Responses as Individual Data Elements—Counting the data elements of a particular function type is necessary to determine the level of complexity for External Input transactions. Counting each error message response as a separate data element could force a Low or Average complexity function to be counted as Average or High complexity, increasing its FP value by up to 50%.

Variant 5: Counting Menus as an External Inquiry—CPM 3.0 guidance is clear that navigational menus are not counted as individual function types, but their existence is a factor in increasing the FP complexity adjustment factor. Petitions to the CPC have indicated that a) users see real value in menus, b) that systems are employing more and more menuing capability, and c) that creating menuing structures is consuming more development time. Variants 6 and 7 indicate alternate counting approaches that were in use by the survey respondents.

Variant 6: Counting Menus as one External Inquiry for Each Layer of Menu—See Variant 5.

Variant 7: Counting Menus as one External Inquiry for each Menu Screen—See Variant 5.

Variant 8: Counting Help Screens as Individual Function Types—The CPM 3.0 counting rules state that help screens are counted as External Inquiry function types, and that there

TABLE III
BASE COUNT FOR CASE A

	BASE FUNCTION POINT CLASSIFICATION			
Definition:	Low	Avg.	High	Total
External Input	11 ×3= 33	0 ×4= 0	0 ×6= 0	33
External Inquiry	4 ×3= 12	6 ×4= 24	16 ×6= 96	132
External Output	14 ×4= 56	5 ×5= 25	1 ×7= 7	88
Log. Internal File	18 ×7= 126	0 ×10= 0	0 ×15= 0	126
Ext. Interface File	0 ×5= 0	0 ×7= 0	0 ×10= 0	0
Total Unadjusted Function Points:				379

TABLE IV
PHASEII: CASE STUDY A RESULTS

	Site A	
Variant	FP	% Δ
Base Count	379	
1. Backup Files as Logical Internal Files	484	28%
2. Backup Files as External Output Types	451	19%
3. Count Add/Chg/Del for External Output Types	355	−6%
4. Count Error Message Responses	379	0%
5. Count Each Menu Screen	391	3%
6. Count Each Layer of Menu Structure	385	2%
7. Count a suite of menus as one Query Type	382	1%
8. Count each separate Help Screen	403	6%
9. Count Hard Coded Tables as Logical Internal Files	dna	——
10. Logical Internal File used as transactions for another system	379	0%
11. Count External Interface Files as External Input Transactions	439	16%

is one External Inquiry type for each "calling screen." In the survey, many of the respondents reported that they count one External Inquiry type for the entire suite of help capability, while others count each help screen combination as a separate External Inquiry Type. This variation could be significant in the overall count for a system with substantial help capabilities.

Variant 9: Counting "Hard Coded" Data Tables as Logical Internal Files—The CPM 3.0 does not currently have an official standard in this area. One view is that all files, whether "hard coded" or not should be counted as function types. Another view is that unless the files are "user maintainable" that they should not be counted. If there were sufficient numbers of "hard coded" tables, the FP count could be significantly affected, as Logical Internal Files are heavily weighted in FP counting.

Variant 10: Logical Internal File Used as Transactions for Another System—This variant of rule interpretation and the following one had a great diversity of responses. Both have to do with the ways in which two systems interface with one another. One view is that files which are accessed for purposes other than just information reference purposes should be counted as transactions in one or the other system. The difficulty is centered around the definition of the logical transaction (External Input or External Output) which is (or is not) taking place, and whether it should be counted in one system or the other.

Variant 11: Counting External Interface Files as External Inputs When Used as Transactions—See Variant 10.

4.2. Case Study Results

Each of the three cases is discussed individually below. For each of the cases there are two analysis tables: one containing the base FP count (based on CPM 3.0), and one with a variance analysis summary. A summary of the results of all three cases appears in Table IX. In each of these cases, the effect on the system FP count of each variation on the standard count was evaluated. It should be noted, however, that the total unadjusted FP count of an individual case could be affected by a combination of application of the rules, which might result in cumulative variations which exceed any one of the individual variances from the application of a single rule change. A worst case analysis will be presented after the presentation of the main results.

4.2.1. Site A Results This case was provided by a Fortune 100 diversified manufacturing and financial services company. This accounting application supports the need for rapid access to information from a variety of separate Accounts Payable applications. It was designed to operate in a PC/LAN environment, and is primarily used by accountants for inquiry purposes. It has built-in help facilities which can be maintained by the users of the system. The base size for the system analyzed at site A was 379 unadjusted FP_s.[5] The system had a wide range of function types developed under a relational data base technology. The users had an exceptionally high degree of interaction with the design and development team, and worked with them to develop and document the system. The documentation for this system was the most extensive of all the cases which were investigated. The functionality of the system does not demand a robust, multi-tiered menu system, but the users did require extensive "Help" capabilities. These capabilities allow the users to continue to update the "Help" screens as required by changes in business practice or better understanding of the assistance which the users of the system need. The error messages of the system were also highlighted using color and emphasized text. In the evaluation of complexity factors, the system rated high marks for its design for End User Efficiency.

The results of the FP variance analysis are presented in Table IV. Three of the variants (1, 2, and 11) produced significant variances in the count, where a variance is deemed to be significant if it is larger than the 12% total difference observed in previous research.

[5] As an aside, the original count done by the developers at site A (not the base count shown in Table III) was the only case which did not comply with all of the counting rules as contained in Release 3.0 of the CPM. The original count was 418 FP's, which is 10% higher than the value achieved through application of the CPM 3.0. This is additional evidence of the need for the current study, and for the further promulgation of counting standards.

TABLE V
BASE COUNT FOR CASE B

	BASE FUNCTION POINT CLASSIFICATION			
Definition:	Low	Avg.	High	Total
External Input	41 ×3= 123	1 ×4= 4	4 ×6= 24	151
External Inquiry	0 ×3= 0	7 ×4= 28	4 ×6= 24	52
External Output	1 ×4= 4	10 ×5= 50	1 ×7= 7	61
Log. Internal File	13 ×7= 91	0 ×10= 0	2 ×15= 30	121
Ext. Interface File	0 ×5= 0	0 ×7= 0	0 ×10= 0	0
Total Unadjusted Function Points:				385

TABLE VI
PHASE II: CASE STUDY B RESULTS

		Site B
Variants	FP	% Δ
Base Count	385	
1. Backup Files as Logical Internal Files	506	31%
2. Backup Files as External Output Types	451	17%
3. Count Add/Chg/Del for External Output Types	385	0%
4. Count Error Message Responses	385	0%
5. Count Each Menu Screen	427	11%
6. Count Each Layer of Menu Structure	394	2%
7. Count a suite of menus as one Query Type	388	1%
8. Count each separate Help Screen	385	0%
9. Count Hard Coded Tables as Logical Internal Files	dna	——
10. Logical Internal File used as transactions for another system	385	0%
11. Count External Interface Files as External Input Transactions	385	0%

Three of the variant analyses merit further explanation. Variant 3 shows a negative variance from the base count. This is a result of the particular counting rule, which allows the counting of Add/Change /Delete transactions as separate function types. Failure to apply the rule properly reduces the FP count. In all other variants, the rule prohibits counting particular function types. These variants result in positive variations from the base count.

Variant 9 is recorded as "dna"—data not available. No hard coded tables were noted in the documentation, but as there was no access to the source code to confirm this result it has been conservatively recorded as "dna."

The calculation for the impact of Variant 11 required making an assumption concerning the level of complexity of the transactions related to the "external interface files." The analysis assumes that the 15 associated transactions were of average complexity, resulting in a variance of 16%. The maximum impact (if they were of "high" complexity) would have been 21%, and the minimum impact (if they were of "low" complexity) would have been 11%.

4.2.2. Site B Results: This case was provided by a Fortune 50 diversified financial services organization that has recently implemented a quality improvement program within their information systems organization. A major part of that commitment is the measurement of various aspects of the systems development process. The subject system is the focal point of this measurement process. It is the data base for all the measurement data which is being collected within the several divisions of the company. The system was designed and implemented by the primary user, who is a member of the IS community.

The system under study was developed as a stand-alone PC application, using a relational data base technology. The application is initially used by a single individual, but is expected to be expanded in its availability as its data bases become more robust. The application supports the management of the development function of the business, providing data and analysis to the managers of the software development and maintenance functions. The system was designed with a robust set of menus to give the users easy access to the data. This system had an unadjusted FP count of 385 points, the largest of those studied. It was a well-documented system and the counts for the system followed the CPM 3.0 guidelines precisely.

This system was the most heavily "menued" of the systems studied, but did not have any "Help" capabilities. The system used extensive relational files, but did not have any External Interface Files. The results of the variant analysis are shown in Table VI.

Variant 3 applies specifically to the existence of Add/Change/Delete output transactions. This system did not have separate transactions associated with outputs, therefore resulting in no change to the total number of FP's. As an aside, however, the subject system did have a wide range of A/C/D transactions associated with the External Inputs to the system, all of which were enumerated as individual function types. They are included in the base count. If these function type triples had been counted only once, there would have been a reduction of 96 FP's, or 24%.

4.2.3. Site C Results: This case was provided by the high technology division of a Fortune 100 aerospace manufacturing company. The system is used to track information concerning the management of various "programs" that are in process within the division. The system specifically tracks the backgrounds of the program managers. It was written in a fourth generation language, and operates on a large central computer, which is accessible from networks of PC's and terminals. It has a simple menu structure, and contains no help capabilities. This system was the smallest of the systems studied for the purposes of this research, with an unadjusted FP count of 208 FP's. This system had few menus, no specific External Outputs, but was dominated by External Inputs and Inquiries. It was designed quickly to fulfill a very specific need, utilizing a fourth generation language and relational data base tool. This data base and the system were originally created for the responses to a "Program Manager Questionnaire." The system provides both pre-programmed inquiries and reports as well as the capability for *ad hoc* inquiries.

TABLE VII
BASE COUNT FOR CASE C

	FUNCTION POINT CLASSIFICATION			
Definition:	Low	Avg.	High	Total
External Input	17 ×3= 51	2 ×4= 8	0 ×6= 0	59
External Inquiry	16 ×3= 48	8 ×4= 32	0 ×6= 0	80
External Output	0 ×4= 0	0 ×5= 0	0 ×7= 0	0
Log. Internal File	9 ×7= 63	0 ×10= 0	0 ×15= 0	63
Ext. Interface File	1 ×5= 5	0 ×7= 0	0 ×10= 0	5
			Total Unadjusted Function Points:	207

The documentation for this system was primarily the source code and the FP calculations. Since this was the only case study in which the authors had access to the source code, it was the only one in which a determination about "hard coded tables" could be made with certainty. The code revealed no "hard coded tables" which might have been counted as Logical Internal Files.

These results shown in Table VIII are quite similar to those for case B, with variants number 1 and 2 showing significant differences, while the remainder had minor or no effect. The main contrast with case B is in variant number 5, where the effect of counting each menu screen was much less pronounced in case C than in case B.

4.2.4. Summary of Results Table IX summarizes the results for all three case studies. Fig. 1 presents the average impact in graphic form.

The remainder of this section categorizes and discusses each of the variants as follows. Variants that resulted in greater than 10% differences in counts in *all* three cases are termed *consistent* sources of variation. Remaining variants that resulted in greater than 10% differences in at least one of the three counts are termed *likely* sources of variation. Remaining variants that resulted in greater than 5% differences in at least one of the counts are termed *possible* sources of variation, while all other variants are termed *unlikely* sources of variation.

4.2.4.1. Topics identified as consistent sources of variation: For a variant to be identified as a consistent source of significant variation it needed to generate more than a 10% difference in reliability in all three cases. Only one survey-identified variant met this criteria: Counting File Backups. The most consistent variation in counts occurred in the area of counting backup files due, in part, to the fact that the Logical Internal Files have the highest individual FP weights. The impact of the differences in the counts was surprising. If backup files are counted, the cases identified an impact of 17% to 31% variation, the largest single source of variance. The lowest variability was observed in the case where backup files were counted as External Output types and the highest in the case where they were counted as additional Logical Internal Files.

TABLE VIII
PHASE II: CASE STUDY C RESULTS

	Site C	
Variants	FP	% Δ
Base Count	208	
1. Backup Files as Logical Internal Files	271	30%
2. Backup Files as External Output Types	244	17%
3. Count Add/Chg/Del for External Output Types	208	0%
4. Count Error Message Responses	208	0%
5. Count Each Menu Screen	214	3%
6. Count Each Layer of Menu Structure	214	3%
7. Count a suite of menus as one Query Type	208	0%
8. Count each separate Help Screen	208	0%
9. Count Hard Coded Tables as Logical Internal Files	208	0%
10. Logical Internal File used as transactions for another system	208	0%
11. Count External Interface Files as External Input Transactions	208	0%

4.2.4.2. Topics identified as likely sources of variation: For a variant to be identified as a likely source of significant variation it needed to generate more than a 10% difference in reliability in at least one of the three cases. Two survey-identified variants met this criteria.

- Counting Menus—In two of the cases, counting (or not counting) menus had an insignificant impact on the total FP count (3%). In one case, where the system was heavily supported by a set of menus the impact was more substantial (11%). This variation is sufficient to introduce a single source of variability which is worthy of further analysis.

  A future related concern is that as Graphical User Interfaces (GUI's) become more widespread, users may demand more robust menuing capabilities. As menuing becomes the rule, rather than the exception, issues surrounding the counting of menus may become more significant in terms of their impact on the reliability of FP counts.
- Counting External Interface File Transactions—Two of the systems had interfaces to other systems. However, the variant was only observed in one of the systems studied. The other case (Site C) used an External Interface File strictly for reference purposes, and not to update a data base. The overall impact was below the threshold of 10%, but the single case in which it applied caused a 16% variation in count. The highest percentage of respondents to the survey (36%) indicated that they would count the transactions. The IFPUG CPC has taken a clear position on counting these transactions, yet there is significant diversity in adherence to the rules. These results further indicate the need to communicate the counting rules and to reinforce the need for consistency.

4.2.4.3. Topics identified as possible sources of variation: The following variants resulted in 5% or greater variance in at least one case.

TABLE IX
PHASE II: CASE STUDY RESULTS SUMMARY

Cases	Site A		Site B		Site C		Mean
Variants	FP	%Δ	FP	%Δ	FP	%Δ	%Δ
Base Count	379		385		208		
1. Backup Files as Logical Int. Files	484	28%	506	31%	271	30%	29.7%
2. Backup Files as External Output Types	451	19%	451	17%	244	17%	17.7%
3. Count A/C/D for External Output Types	355	-6%	385	0%	208	0%	−2.0%
4. Count Error Message Responses	379	0%	385	0%	208	0%	0%
5. Count Each Menu Screen	391	3%	427	11%	214	3%	5.7%
6. Count Each Menu Structure Layer	385	2%	394	2%	214	3%	2.3%
7. Count a suite of menus as one Query Type	382	1%	388	1%	208	0%	0.7%
8. Count each separate Help Screen	403	6%	385	0%	208	0%	2.0%
9. Count Hard Coded Tables as Logical Internal Files	dna	—	dna	—	208	0%	0%
10. Logical Internal File used as transactions for another system	379	0%	385	0%	208	0%	0%
11. Count External Interface Files as External Input Transactions	439	16%	385	0%	208	0%	5.3%

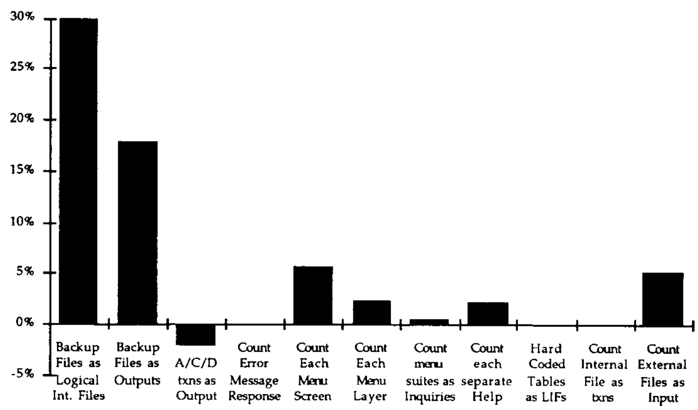


Fig. 1. Average percentage change in function point count by variant.

- Counting Add/Change/Delete Transactions—The question stated in the survey focused on the counting of External Outputs from A/C/D transactions. Only one of the case study examples identified individual outputs from the A/C/D transaction sources. In this case, there was a variance in the total count of 6%. In two of the cases, the FP counts included separate counts of A/C/D input transactions. This is compliant with CPM 3.0 guidance. However, if there were only one External Input function counted for each of the A/C/D triples, there would have been a 25% reduction in overall FP counting one case, and a 10% reduction in the other. Again, these are substantial variations in the overall FP counts, which could have a significant detrimental impact on reliability.
- Counting Help Screens—Only one of the systems contained a "Help Facility." In this case changes in the application of the counting rules resulted in a 6% overall change in the FP count. This variation, while smaller than the impact of backup files, still merits some concern. Users are increasingly requiring internally built systems to match the functionality of off-the-shelf software, which is typically equipped with Help and other facilities. It is reasonable to expect that these functions will account for more of the overall functionality of systems in the future. In this regard, a current 6% variation due to this rule interpretation is one which may demand further consideration.

4.2.4.4. Topics identified as unlikely sources of variation: These survey-identified variants tended to result in small or zero bottom line variances.

- Counting Error Message Responses—None of the cases studied had error messages associated with External Input transactions. This is the only case in which CPM 3.0

allows the counting of error messages. In the one case (Site A) in which the error messages were present, they were only associated with inquiries. Even if the counting rule were to be applied to the inquiries there was very little variation. Of the ten transactions (inquiries) which were potentially affected, most were already classified as high complexity. These inquiries already had achieved the highest point value available, and counting any additional data elements could not have raised the point score. Only the three of these transactions that were classified as average could have been affected in a recount. The analysis would have increased their point value from 4 to 6 points each, increasing the total FP count for the system by 6 points, or 1%. This may be indicative of the small impact to be expected through this variant.

- Counting Menu Structures and suites—There were three menu variants analyzed for their impact on the overall count. Only one of these (counting each screen, discussed above) had the potential of making a substantial impact on the overall FP count. The other variants had very small impact.
- Counting Files used by Other Systems as Transactions—None of the three cases that were reviewed contained evidence of Internal Logical Files which were used by other systems as Input Types. The case studies were restricted to recently developed systems, and it is possible that one or all of these systems may have an Internal Logical File used as an External Input to another system in the future. The variant may generate different results in that circumstance, but was not found to have any effect on the systems examined.

4.2.4.5. Topics not able to be identified as to their likelihood as a source of variation: Counting Hard Coded Tables—The source code necessary to investigate this feature was only available at site "C" where it was determined that no hard coded tables existed, and hence the impact of counting variants was zero. Clearly, this result should be interpreted especially cautiously, since it may be an artifact of this particular site.

4.2.4.6. Worst case analysis: In all of the above analyses, each variant was analyzed separately in order to identify those variants that most merited management attention. An additional question is what if a site were to be unfortunate enough to have chosen every variant that would maximize the difference between its FP count and the count achieved by following standard practice? Note that this difference is not simply the sum of the 11 variants, as not all of the variants are independent. For example, variants 1 and 2 are two different means of treating backup files. A site could choose one or the other instead of the standard, but could not logically choose both. Specifically, the maximum positive variance scores shown in Table X are determined as the summation of the percentage variance from variants 1, 5, 8, and 11.

It should be emphasized that the average worst case result of 43% is not inconsistent with previous research reporting variance among counters of approximately 12%. This is because the previous research focused on typical or average case behavior, whereas the larger figure represents the largest possible variance at these sites given the choice of those variants previously identified as contentious areas and deliberately choosing all of those which would create the largest arithmetic difference.

TABLE X
WORST CASE RESULTS

Site	Maximum Negative Variance	Maximum Positive Variance
A	−6%	53%
B	0%	42%
C	0%	33%
Average	−2%	43%

4.3. Implications of the Results

4.3.1. General Results: A general interpretation that may be gleaned from this research when it is taken in conjunction with other recent research on the topic is that FP's are a more reliable metric than is casually believed. A particular contribution of the current paper to this general result stems from the deliberate research design to investigate variations in FP counting practices that were *a priori* believed to be important sources of significant variation, but which were found, with the exception of the treatment of backup files, to generally have only a minor influence on the final count for the systems studied. While the case studies specifically addressed only a small number of systems, these general results are consistent with recent previous work that examined average behavior for a larger number of systems. Therefore, these combined results should be encouraging both to organizations that have already adopted FP's, and for organizations that are currently considering their adoption.

However, beyond this general result there are clearly areas in which the definition of FP's could be improved. Most important among these is the proper counting of backup files. IFPUG and any other standard setting organizations in this area need to adopt and promulgate a clear and consistent rule on this topic, as this is the area that was identified in the research as posing the greatest threat to counting reliability.

4.3.2. Implications for Standards Settings: There is a need to act on the findings of this research. Standard setting bodies should take a series of actions to improve the reliability of FP counts.

- Identify and resolve outstanding and contentious issues—Even after the specific issues addressed in this research are resolved, the rapid pace of change in information technology virtually guarantees that new issues will arise. To address this issue, a regular approach by a standards setting body needs to be put into place to institutionalize the type of research presented here. This research would consist of two phases, the first an identification phase to identify potential problem areas, and a case study phase where the effect of these potential problems is assessed. Without such a process in place it is likely that FP counting standards are likely to significantly lag actual practice.

- Communicate standards for issues of frequent variation—A special communication should be prepared to emphasize the need for consistent application of existing counting rules. This conclusion is underscored by the noncompliance results shown in the survey.
- Continue research into areas of potential variability—There are other areas of variability which will become more prominent in the future. There must be a continuing program of research to insure that these areas are identified and counting standards written.

The need for greater communication of existing standards is readily apparent from the data in Table I. The results of a survey of leading FP measurers demonstrate that for three issues, Error Messages, Menu Function Types, and Menu Function Count, the majority answer was not the CPM 3.0 standard. This indicates a need for greater communication of the CPM 3.0 results to the membership.[6] The survey also revealed issues, such as External Inquiry function weighting, for which no additional special effort is deemed necessary.

4.3.3. Implications for Organizations Counting FP's: Consistent counting of FP's within an organization is of extreme importance. It provides the basis for comparison of systems measures across system, departments, and locations. This consistency can be gained by creating ones own standards, or by adopting the standards of others. The results of the research and the case studies indicate that organizations that adopt the CPM 3.0 standards do count reliably. Its adoption can provide a firm basis for consistent counting, and, like all industry standards, is likely to be updated to reflect contemporary issues in counting.

In both cases where the organizations were trained using the CPM 3.0, the base count was in compliance with the counting practices. In the case where the organization had been trained in counting FP before the publication of CPM 3.0 there were significant deviations from the CPM 3.0.

Measurement is the means by which management knows that objectives are being met. The accuracy of these measures over time, and across various systems, organizations, and even companies, is an essential component to appropriate decision making. Through this and related research FP's have been shown to be a reliable measurement instrument. Managers should adopt them as a measure of system size, and follow and endorsed standard in their use. FP's are the only measure supported by an independent standards setting body, with an established issue resolution process. It is this standardization which will continue to improve the ability of FP to measure system size. This improvement requires the active cooperation of organizations which are using FP-based measures in identifying potential sources of variation, and suggesting solutions to the standard setting body.

[6] Since this survey was completed, the CPC has continued to release newer versions of the CPM.

4.3.4. Implications for Automation of FP's: In order to reduce the cost of FP counting, and to reduce human error, a number of proposals have been made to automate FP counting, via either stand-alone tools or as embedded within CASE technology. A critical precursor to successful automation of FP counting is a clear set of well-defined measurement conventions. The current research results have three implications for the automation of FP counting. The first is the obvious need for the tools to carefully define their counting conventions, given the potential impact of adopting nonstandard variants. Second, the tools should be able to communicate these conventions to the user. Failure to do so may lead to unsuccessful adoption of the tool by organizations that have previously been counting FP's manually. If, for example, a tool has adopted significantly different conventions than those used at the site, then initial benchmarking of the tool by experienced users may come to the conclusion that the tool is inaccurate, when, in fact, it may be merely consistently applying variant counting conventions. Finally, a suggestion for tool vendors arising from these results is to provide some sensitivity analysis as part of the output of the tool. For example, following the variance approach taken in this research, the tool could produce as output both its standard count plus some alternative counts based on differing assumptions. This could also highlight for users the features of the application which are most significant in driving the final count, which in turn might be a useful planning tool for project managers.

V. Concluding Remarks and Future Research

This paper identifies the source and impact of variations in the application of FP counting rules. The results of this analysis should provide guidance to FP standard setting bodies in their deliberations upon rule clarification, and to practitioners as to where the difficulties lie in the current implementation of FP's. In turn, the result of this effort should continue the process of improving the quality and reliability of measures of software size, productivity, and quality.

FP's currently provide the only established industry standard of size measurement in the area of systems development. The measurement of productivity also requires equivalent standardization of resource (cost and time) measurement. Few organizations have the same rules for accounting for staff time applied to projects. If there is to be further comparison of measurement across companies, and the development of more refined estimating capabilities, standards will need to be established in a wide variety of areas of software development management. Some recent work by the IEEE Software Productivity Metrics Working Group of the Software Engineering Standards Subcommittee is a step in this direction, as is work by the Software Engineering Institute at Carnegie Mellon University.

The issues upon which this research have focused center on the clarification of counting guidelines for systems that are traditional in nature. The object is to refine the counting guidelines, and to drive out the ambiguity of current measurement conventions. This is an issue of considerable

TABLE XI
COMPLEXITY ASSIGNMENT FOR EXTERNAL OUTPUTS [24]

	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 XII
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

practical importance, since there are so many systems for which these measures are relevant. However, the issue of measurement reliability is much larger than just the issues outlined within the context of this research. The advent of event driven, object oriented systems; knowledge based systems; plus real-time and scientific systems may require redefinition of FP's or the development of one or several new measures to identify system size. For example, an initial set of metrics for object-oriented design has been proposed [8].

Systems development is an intellectual activity, the conversion of an idea into software. However, if the IS profession is to improve the way in which this critical work is done then measurement of this intellectual activity is necessary. Perfect measures may never be developed, but efforts directed toward this goal should result in improved metrics and therefore wider adoption in practice. Improving the quality of this one measure, FP's, is but a start in the effort to improve management's ability to measure all the aspects of software development and maintenance. Objectives of managers today include productivity and quality, but are certainly not limited to them. Increased efforts to improve the reliability of these measures will continue to enhance their acceptance and credibility in both the worlds of the systems professionals and general management.

APPENDIX 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 [24]. The following is a minimal description only. Calculation of FP's 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 XI 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 FP's 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 XII. Then, the total number of function counts (FC's) is computed as shown in Equation (1).

$$\text{FC} = \sum_{i=1}^{5}\sum_{j=1}^{3} w_{ij}x_{ij} \tag{1}$$

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 then summed and modified to compute the value adjustment factor (VAF):

$$\text{VAF} = .65 + .01\sum_{i=1}^{14} c_i \tag{2}$$

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 FP's:

$$\text{FP} = \text{FC}(\text{VAF}). \tag{3}$$

ACKNOWLEDGMENT

The cooperation of A. Belden, M. Braun, and J. Frisbie was invaluable in providing data for this research. Helpful comments were received from J.M. Desharnais, F. Mazzucco, J. Quillard, R. Selby, C. Scates, W. Rumpf, and L. Smith on an earlier version.

In this section, we would like you to answer the questions using your organization's 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 "Multi-Function 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): ______

Multi-Function Address Screen

Name:____
Address:______
City:________
State: _ Zip___

transaction confirmation message goes here

PF1 = Add PF2 = Change PF3 = Delete

Fig. 2. FP counting practices survey.

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

All Possible Error Messages (20 in total)
1. Name too long.
2. Name too short.
3. Not a valid city.
4. Not 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

Screen Layout Hierarchy

```
                    |---- Manage Inventory ----
                    |---- Plan Acquisition -------
Main Menu    -----  |---- Update Catalogue ------
                    |---- Support Inquiries ------
                    |---- Produce Reports --------
```

Fig. 2. FP counting practices survey *(continued)*.

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: ______

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

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.

Fig. 2. FP counting practices survey *(continued)*.

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 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):
______ 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): ______

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 dates 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): ______

Fig. 2. FP counting practices survey *(continued)*.

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

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 an 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):

Fig. 2. FP counting practices survey *(continued)*.

REFERENCES

[1] A.J. Albrecht in *IFPUG Function Point Counting Practices Manual*, J. Sprouls, Ed. Westerville, OH: International Function Point Users Group, 1990.

[2] A.J. Albrecht and J. Gaffney, "Software function, source lines of code, and development effort prediction: A software science validation," *IEEE Trans. Software Eng.*, vol. SE-9, pp. 639–648, 1983.

[3] R.D. Banker *et al.*, "A model to evaluate variables impacting productivity on software maintenance projects," *Management Science*, vol. 37, pp. 1–18, Jan. 1991.

[4] R.D. Banker and C.F. Kemerer, "Scale economies in new software development," *IEEE Trans. Software Eng.*, vol. SE-15, pp. 416–429, Oct. 1989.

[5] C.A. Behrens, "Measuring the productivity of computer systems development activities with function points," *IEEE Trans. Software Eng.*, vol. SE-9, pp. 648–652, Nov. 1983.

[6] D.B. Bock and R. Klepper, "FP-S: A simplified function point counting method," Southern Illinois Univ., Edwardsville, working paper, Oct. 12, 1990.

[7] E.G. Carmines and R.A. Zeller, *Reliability and Validity Assessment.* Beverly Hills, CA: Sage Publications, 1979.

[8] S.R. Chidamber and C.F. Kemerer, "Toward a metrics suite for object oriented design," in *Proc. 6th ACM Conf. Object Oriented Programming, Systems, Languages and Applications (OOPSLA)*, 1991.

[9] J. Cooprider and J. Henderson, "A multi-dimensional approach to performance evaluation for I/S development," MIT Center for Information Systems Research, working paper, no. 197, Dec. 1989.

[10] J.-M. Desharnais, "Analyse statistique de la productivite des projets de developpement en informatique a partir de la technique des points de fonction (English version)," Universite du Quebec a Montreal, masters thesis, 1988.

[11] J.B. Dreger, *Function Point Analysis.* Englewood Cliffs, NJ: Prentice Hall, 1989.

[12] R.D. Emrick, "Software development productivity second industry study," in *Proc. Conf. Int. Function Point Users Group*, pp. 1–44, 1988.

[13] R. Jeffery, Personal communication, Aug. 15, 1990.

[14] C. Jones, *Programming Productivity.* New York: McGraw-Hill, 1986.

[15] ——, "A short history of function points and feature points," Software Productivity Research, Inc., mimeo version 2.0, Feb. 20, 1988.

[16] C.F. Kemerer, "An empirical validation of software cost estimation models," *Commun. ACM,* vol. 30, pp. 416–429, May 1987.

[17] ——, "Reliability of function points measurement: A field experiment," *Commun. ACM,* to appear in Feb. 1993.

[18] G.C. Low and D.R. Jeffery, "Function points in the estimation and evaluation of the software process," *IEEE Trans. Software Eng.*, vol. 16, pp. 64–71, Jan. 1990.

[19] W.E. Perry, "The best measures for measuring data processing quality and productivity," *Quality Assurance Institute Tech. Rep.*, 1986.

[20] R.S. Pressman, *Software Engineering: A Practitioners Approach.* New York: McGraw-Hill, 1987.

[21] D.J. Reifer, "Asset-R: A function point sizing tool for scientific and real-time systems," *J. Syst. Software,* vol. 11, pp. 159–171, 1990.

[22] E.E. Rudolph, "Productivity in computer application development," Univ. Auckland, Dept. of Management Studies, Auckland, New Zealand Working Paper 9, ISSN 0110–4756, Mar. 1983.

[23] C. Scates, Personal communication, Aug. 5, 1991.

[24] J. Sprouls, *IFPUG Function Point Counting Practices Manual*, Release 3.0, International Function Point Users Group, Westerville, OH, 1990.

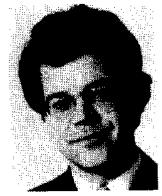
[25] E.B. Swanson and C.M. Beath, "The use of case study data in software management research," *J. Syst. Software*, vol. 8, pp. 63–71, 1988.

[26] C.R. Symons, "Function point analysis: Difficulties and improvements," *IEEE Trans. Software Eng.*, vol. 14, pp. 2–11, Jan. 1988.

[27] A. Topper, "CASE: A peek at commercial developers uncovers some clues to the mystery," *Computerworld*, vol. XXIV, pp. 61–64, Apr. 9, 1990.

[28] S. Whitmire *et al.*, "Scientific and real-time software size metrics—Draft preliminary technical report," The Boeing Company, Jan. 9 1991.

[29] K. Zwanzig, *Handbook for Estimating Using Function Points*, GUIDE International, Project DP-1234, 1984.



Chris F. Kemerer received the B.S. degree in decision sciences and economics from the Wharton School of the University of Pennsylvania and the Ph.D. degree from the Graduate School of Industrial Administration at Carnegie Mellon University.

He is currently 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 for improved performance, and he has previously published articles on these topics in *Communications of the ACM, Information and Software Technology, Management Science, IEEE Software,* and the IEEE Transactions on Software Engineering. He serves on the editorial boards of the *Communications of the ACM, Information Systems Research, Journal of Organizational Computing, Journal of Software Quality,* and *MIS Quarterly.*

Dr. Kemerer is a member of the ACM and The Institute for Management Sciences.

Benjamin S. Porter received the B.S. degree in mathematics from Carnegie Mellon University and the M.S. degree in industrial administration from the Krannert Graduate School of Industrial Administration at Purdue University.

He is a manager in the Information and Technology Strategy practice of Andersen Consulting, and heads their research programs in information and technology strategy. His research interests are in the measurement of information systems processes and linking the performance to the overall performance of the business enterprise. He has served on the Board of Directors of the International Function Point Users Group as chairman of the first Counting Practices Committee, and has spoken widely on the subject of information technology management and measurement.