

AD 680 105



SP-3122

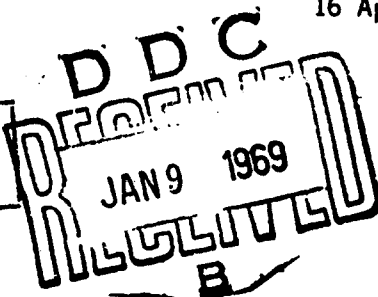


**Some Recent Contributions to
Computer Programming Management**

Edward A. Nelson

16 April 1968

This document has been approved
for publication in its
entirety.



SP *a professional paper*

Some Recent Contributions
to
Computer Programming Management

Edward A. Nelson

16 April 1968

SYSTEM
DEVELOPMENT
CORPORATION
2500 COLORADO AVE.
SANTA MONICA
CALIFORNIA
90406

This paper was prepared for presentation at
the 1968 meeting of the Association for
Computing Machinery in Las Vegas.



ABSTRACT

SOME RECENT CONTRIBUTIONS TO COMPUTER PROGRAMMING MANAGEMENT

The System Development Corporation (SDC) has been conducting research into the management of the computer programming process since 1963. This paper briefly describes some of the results of this research, plus individual research by the author, and the relevance and usefulness of the results to the operating manager.

The paper emphasizes four specific types of management tools that were produced: (1) planning aids; (2) cost estimating guides; (3) a project reporting and control system; (4) a technique for evaluating the effectiveness of certain classes of computer-centered information systems. Each of these tools is briefly described, and the research design and procedures used in their development are mentioned. The experience gained -- at SDC and elsewhere -- in applying traditional research techniques to the computer programming process has yielded certain insights regarding the economic and management dimensions of computer programming, and several of these insights are discussed. Foremost among this author's conclusions is that the management principles that apply to any other coordinative activity are equally applicable to the computer programming process. Specific techniques may differ, especially at the lower echelons, but these differences pertain to the technical skills and procedures of the production process; they are engineering, not management or economic issues. The paper concludes with several general observations pertinent to future research in computer programming economics and management, and the use of the management tools described.

SOME RECENT CONTRIBUTIONS
TO COMPUTER PROGRAMMING MANAGEMENT

INTRODUCTION

This paper will discuss four comparatively recent tools that can be useful to the manager of computer programming: (1) a planning aid; (2) cost estimating guides; (3) a project reporting and control system; and (4) a technique for evaluating the effectiveness of certain classes of computer-oriented information systems. The contribution of general management principles, and the degree to which they shape the process of creating tools or techniques, should become apparent in the following sections of this paper.

Management is defined here as the process of accomplishing objectives by establishing an environment favorable to performance by people operating in organized groups. The essence of managing consists in the attainment of coordination or harmony of individual effort toward the achievement of group goals. The management process may be said to consist of the performance of specific functions, namely: planning, organizing, staffing and assembling resources, direction, and control (1). Each of these functions applies also to the management of projects whose products are computer programs.

16 April 1968

3

SP-3122

It is my opinion that principles of management exist that are universal to all forms of coordinative activity. These principles consist of the fundamental causal relationships that explain and help predict the results of the management process, and serve to improve the results of the process in terms of the desired goals. Management principles, therefore, apply equally well to the fighting of a battle, the running of a technical meeting, or the creation of a computer program. One such universal management principle, for example, has been called the "principle of the primary of planning" (1). That is, planning is the primary requisite to the other managerial functions of organizing, staffing, direction, and control. This means that the degree of control over a programming project (regardless of whether we are talking about schedules, man-hours, or other resources) can be no greater than the extent to which adequate plans have been made for the project; it can be less, of course, since contingencies can force modifications in even the best laid plans--but the extent of planning sets the degree of control that is possible. I suspect that this is the primary reason why control is lost on many computer programming projects. It is not the comparative newness of the computer programming process, difficulties with programmers, or technical factors--it is simply that the programming projects are not adequately planned in the first place.

The implications to the managers of computer programming of the universal applicability of management principles is that their immediate and most productive course of action is to apply the principles and tools already

16 April 1968

4

SP-3122

at hand. The magnitude of improvements possible with this approach is illustrated by my own research on administrative costs in the life insurance industry (2), which showed that administrative costs could be at least halved for the average life insurance company using their existing ADP equipment, providing that these companies manage their affairs so as to achieve the performance already demonstrated by their peers. The implications of this universality of management principles to researchers, on the other hand, is that these principles can become the guidelines for useful research. To the extent that management principles are generally applicable, and the management lessons of other disciplines can be used for computer programming projects, the comparative newness of the computer programming process becomes largely irrelevant; and productive research in the economics and management of the programming activity, given this existing foundation of management principles, will most profitably focus on the creation of guides and devices for answering technical, rather than managerial questions. This process, including the research emphasis on technological factors, is illustrated by the descriptions that follow.

A PLANNING AID

As in other activities, planning computer programming projects involves the selection from among alternatives of future courses of action. To promote accurate planning by first-level supervisors, the Office of Naval Research sponsored the development of a planning guide (3), designed to stimulate more complete consideration of the entire computer programming process. A set of planning and management tasks are defined in terms of the computer program development process; this involved dividing the programming process into distinct phases, or steps, and identifying the detailed tasks (36 separate tasks are defined in the guide) within each step. For each task, the information and document inputs required to perform the task were listed, subtasks (there were 4 to 13 for each task in the guide) are defined, the information and documents resulting as output of the task are listed, as are several of the more important factors that affect the costs of performing the tasks. A sample layout for presenting this information is illustrated in Figure 1. What was done in the construction of the planning guide, in essence, was to study the technical process of computer programming, and prepare a check-list for the planner to consider; the planner can then more readily apply planning principles with which he was already familiar to this technical process.

It is noteworthy that the steps in the computer programming process, although containing technical distinctions, are nevertheless quite similar to those in the procurement of other systems. This is illustrated in Figure 2, where a comparison is made between steps in a general system life cycle. The four general steps in Figure 1 are substantially equivalent to the procurement

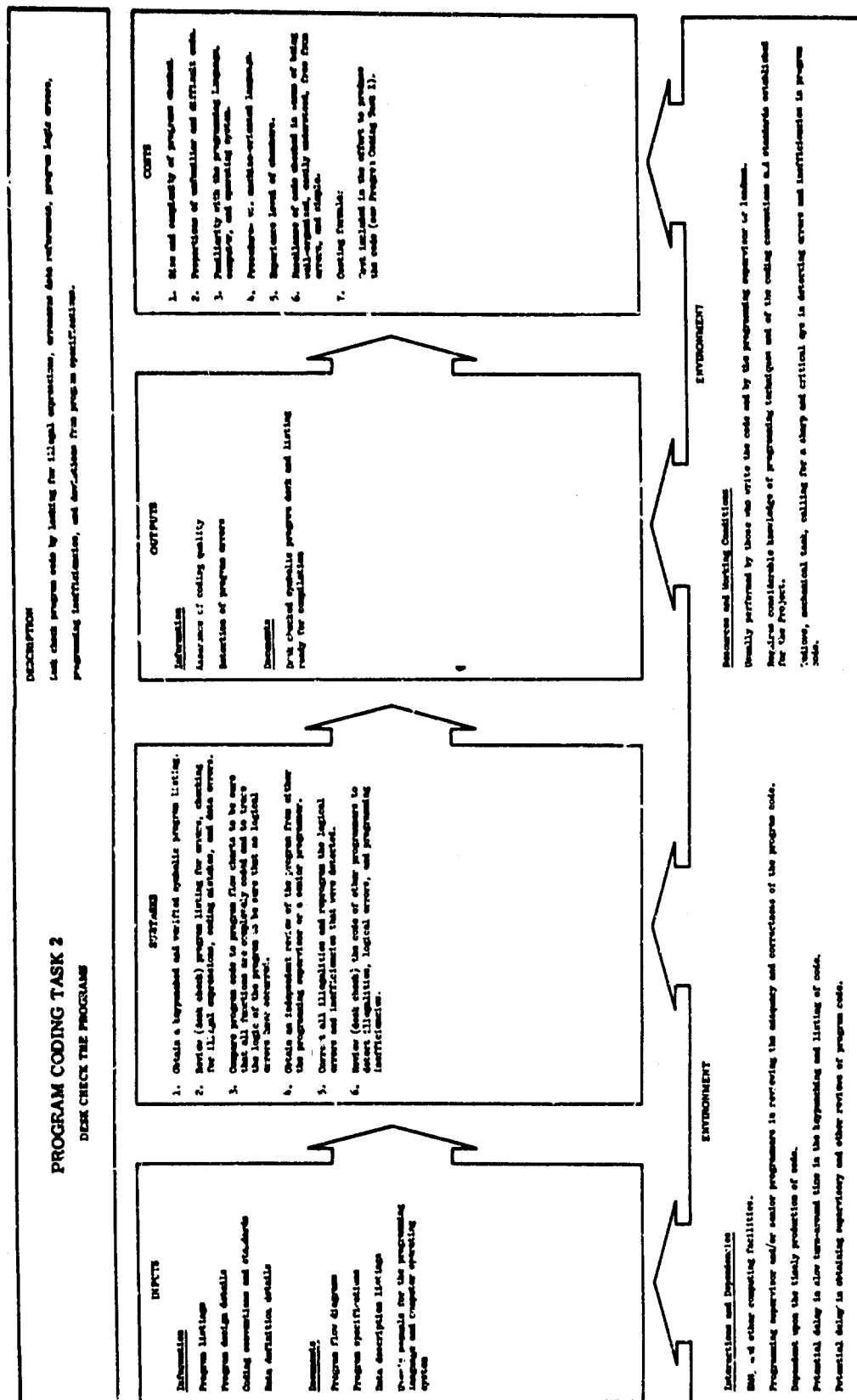


Figure 1. Sample Layout From the Planning Guide

General System Life Cycle	Steps in the Computer Program Life Cycle
Economic Feasibility Study	1. Information Processing Feasibility Analysis
System Definition	2. Information Processing Analysis
	3. Information Processing Design
System Acquisition and Installation	4. Computer Program Design
	5. Computer Program Coding and Checkout
	6. Computer Program Functional Test
	7. Information Processing Integration Test
	8. Information Processing Installation and Implementation
System Operation	9. Information Processing Program Maintenance

Figure 2. Relationship of the Computer Program Procurement Process to the Procurement of Other Products.

16 April 1968

8

SP-3122

phases (conception, definition, acquisition, operation) specified in the USAF system management process (4), which was developed originally for the procurement of weapons systems hardware; in fact, the universality of management principles and their applicability to the computer programming process is again illustrated by the efforts expended in applying the USAF systems management concepts to the management of computer programming (5, 6, 7, 8, 9).

COST ESTIMATING GUIDES

The Electronic Systems Division of the Air Force Systems Command has sponsored work directed toward the analysis of experience data to develop tools for estimating the costs (man hours, computer hours, and other resources expended) of the computer programming process. One of these studies, by the System Development Corporation, collected data by questionnaire on 169 computer programs by both government and industry. This work was conducted in cycles (10), each marked by collection and analysis of new data to improve upon earlier results, and culminated in a handbook (11) that also included a collection of estimating rules-of-thumb gleaned from an examination of the technical literature. The System Development Corporation has been engaged in such research on the economics and management of computer programming since 1963. The second study was done by the Planning Research Corporation, and involved an extensive interview of the cognizant personnel who worked on a total of eighteen computer programs (12). Both of these studies used standard multiple regression techniques to generate equations for relating required resources to the various factors presumed to influence the magnitude of these resources.

The major portion of the statistical material published in the SDC handbook is based on data covering only computer program design, coding, and functional test (steps 4, 5, and 6) of the computer program life cycle described in Figure 2. These data were divided into subsamples based on several types of programs, and some of the characteristics of this subsample data -- in terms of the three basic resources expended in writing programs -- are listed in Table 1. It is interesting to note that in terms of object instructions

TABLE I
RESOURCE EXPENDITURE RATE PER 1000 OBJECT INSTRUCTIONS

TYPE OF COMPUTER PROGRAM	NO. OF DATA POINTS	RESOURCE COST RATE												ELAPSED TIME (MO.)			
		MAX MONTHS $\frac{Y_1}{\bar{Y}_1} \times 10^3$				COMPUTER HOURS $\frac{Y_2}{\bar{Y}_2} \times 10^3$								$\frac{Y_3}{\bar{Y}_3} \times 10^3$			
		MAX	MIN	σ	MEDIAN	MEAN	MAX	MIN	σ	MEDIAN	MEAN	MAX	MIN	σ	MAX	MIN	MEAN
MOI POL	123	100.00	.14	10.18	4.00	5.89	294.04	.05	42.75	15.00	29.52	40.00	.06	5.81	1.33	3.55	
	46	9.49	.07	2.61	1.16	2.13	52.90	.30	13.74	2.86	9.76	18.43	.06	3.11	.92	2.90	
		P < .05															
FORTRAN JOVIAL COMOL OTHER 1	8	9.49	.11	3.88	.97	2.75	50.59	.31	18.02	2.68	10.25	18.43	.17	6.95	1.14	4.50	
	15	7.60	.66	2.31	2.50	3.07	52.90	.77	15.06	16.67	17.73	8.39	.47	1.86	1.18	1.87	
	12	7.33	.07	2.33	.49	1.22	38.33	.30	10.70	1.80	5.25	7.33	.06	1.96	.41	1.00	
	11	4.10	.12	1.57	.57	1.36	14.49	.31	3.86	2.43	3.45	11.27	.08	3.56	.74	2.64	
		N.S.															
BUSINESS SCIENTIFIC UTILITY OTHER	79	38.82	.07	5.11	1.54	3.13	86.00	.23	17.93	3.09	11.25	18.94	.06	4.05	1.11	2.87	
	27	12.00	.14	2.00	3.14	3.55	140.00	.03	28.52	4.83	17.78	18.43	.11	4.28	1.18	2.93	
	28	100.00	.49	18.83	2.28	9.79	294.04	1.06	63.30	38.18	57.36	21.00	.10	4.03	1.03	2.03	
	35	17.65	.66	1.16	4.20	5.89	129.03	3.86	28.42	20.98	30.06	40.00	.24	8.41	2.00	5.14	
		P < .05															
LARGE COMPUTER MEDIUM COMPUTER SMALL COMPUTER	105	100.00	.07	10.51	3.17	5.50	294.04	.05	42.47	12.50	26.72	40.00	.06	5.99	1.00	3.06	
	53	16.67	.12	3.09	2.54	3.19	177.77	.31	30.22	5.20	17.57	18.94	.06	3.92	2.00	3.19	
	11	38.82	.93	10.81	4.00	6.97	80.00	1.09	24.51	34.69	31.14	14.93	.64	4.78	2.98	4.66	
		N.S.															
TOTAL SAMPLE	169	100.00	.07	8.94	2.93	4.87	294.04	.05	38.16	10.44	24.14	40.00	.06	5.34	1.22	3.21	

NOTES

1) P = Probability of erroneously concluding that the population means are different (e.g., $p < .01$ indicates that if one concludes that the population means are different, the probability of error is less than 1 out of 100). If $p > .25$, it is denoted by N.S. (not significant) in this table.

2) All of the planning factors above are based on raw data that has not been winsorized (i.e., extreme values reduced) to prevent distortion by large values.

produced, procedure-oriented languages (POL) demonstrates a statistically significant advantage over machine-oriented languages (MOL), with lower resource expenditure rates for both man-months and computer hours; discussions of this point, and other conclusions from the data, are elaborated elsewhere (11, 13). The data in Table I may be used in estimating resources if a reasonable estimate of the number of object instructions can also be made.

Figure 3 based on a different arrangement of the data, is also useful in estimating man-months when the approximate number of object instructions is known. This curve was developed by arranging all of the programs in the sample in ascending order by production rate (man-months per 1000 instructions), and plotting these production rates for each program against the accumulative percent in the total sample covered by the sequence. Thus, that subset of programs that comprises the 40% of the total sample with the lowest production rates contains, from Figure 3, production rates of two man-months per 1000 object instructions or less. This construction permits the abscissa to serve as an overall measure of the difficulty of the programming job. Example: if the estimator subjectively believed the program to be estimated is more "difficult" than the median of the sample (50% on the abscissa), but not as "difficult" as the more extreme values, he might choose to use production rates for the 60-80 percent range; then the expected resource expenditure rate taken from the ordinate would be 3.9 to 6.3 man-months per 1000 object instructions. In Figure 3, the typical range is arbitrarily defined to exclude the upper and lower 20 percentiles. The high slope of the curve within this typical range is an indication of the large variation in production rates in the sample. This

16 April 1968

12

SP-3122

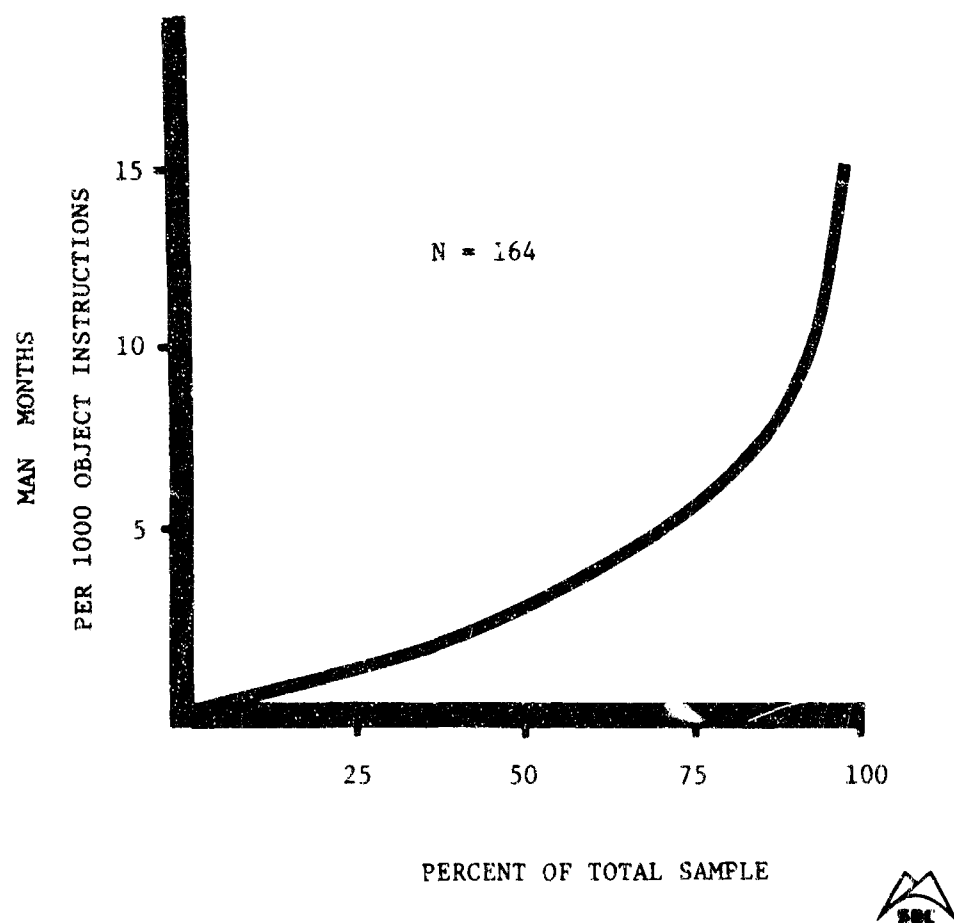


Figure 3. Man Months Expenditure Rates

variation could mean that there are many factors, including intangibles, that affect the expenditure of resources in computer programming; on the other hand, it could also indicate that accepted management principles had not been vigorously applied to the production process of the programs in the sample.

Multiple regression techniques have also been used to investigate the impact of various parameters on the expenditure of resources in computer programming. Figure 4 illustrates one of the outputs of these efforts. Such work not only produces equations, as in Figure 4, that can be used for estimating purposes (with appropriate caveats); it also serves to identify factors that have statistically significant impact on expenditures, and hence directs management attention to those critical factors, many of which are subject to management control. A disadvantage also arises, however, when the results of cost research are published in the form of equations, as in Figure 4. Such equations are frequently interpreted by the user as demonstrating a specific causative relationship; this is not the case. Equations developed by multiple regression techniques do reveal important parameters, and may represent those relationships that provide the most statistically significant manner of describing the character of the analyst's sample; however, they do not necessarily represent natural laws, as do many of the equations used by the engineer or physicist. Also, they must be used in their entirety or not at all: if a value for any one of the independent variables in Figure 4 were not available, the equation could not be used as it stands, since repeating the multiple regression analysis without the missing parameter would result in a reassignment of weights to all of the remaining variables and the Y-intercept of the equation.

DATA BASE:

LARGE COMPUTER SUBSAMPLE (CONSISTING OF SYSTEMS
WHOSE PURCHASE PRICE IS AT LEAST \$750,000.)

N = 106 COMPUTER PROGRAMS

SE = 22

r = .83

r² = .68

MEAN = 54

STD. DEV. = 71

TOTAL MAN MONTHS = Y_1 , WHERE:

$$Y_1 = + .049 + 15.2X_6 - .236X_{25} + .528X_{30} + 4.50X_{37} + .091X_{46} \\ - 17.5X_{48.1} + 25.1X_{51} + 22.0X_{54} + 26.0X_{56} - .251X_{64} \\ - 14.9X_{65} + 10.4X_{74}$$

X_6 - Complexity of Program System Interface. Coded: more than 50% of design effort devoted to data transfer problems to or from the program data point = 2; between 10% and 50% effort to data transfer problems = 1; less than 10% = 0.

X_{25} - Percent Clerical Instructions. Coded in percent.

X_{30} - Percent Information Storage and Retrieval Functions. Coded in percent.

X_{37} - Frequency of Operation. Coded: not applicable = 0; less than 1/month, more than 1/month and less than 1/week = 2; more than 1/week and less than 1/day = 3; daily = 4; utility or on-line (including compilers) = 5.

X_{46} - External Documentation. Coded: number of pages written for, or distributed to, customers.

$X_{48.1}$ - Business. Coded: as mutually exclusive binary variables; i.e., programs classified as business application = 1; remaining applications = 0.

X_{51} - First Program on Computer. Coded: yes = 1; no = 0.

X_{54} - Special Display Equipment. Coded: special display equipment used = 1; not used = 0.

X_{56} - Random Access Device Used. Coded: use of such storage = 1; such storage not used = 0.

X_{64} - Percent Programmers Participating in Program Design.

X_{65} - Personnel Continuity. Coded: number of personnel working for the duration of the project, divided by the maximum number assigned at any one time.

X_{74} - Number of Locations for Program Data Point Development.

Note: Subscripts are those used in original source document (11).

Figure 4. Equation for Estimating Man Months
for Programs Developed on Large Computers

Another application of the available computer programming cost data is illustrated in Figure 5. Here, expenditures of man-months are related to the concomitant expenditures in computer time required to debug the programs in the sample. The comparative advantage of the POLs in this sample of programs is also apparent in these relationships. The relationships in Figure 5, as in the equation in Figure 4, again do not necessarily represent cause and effect; that is, it is not meant to imply that a given expenditure of man-months will require a concomitant indicated expenditure of computer hours. However, to the extent that the sample used in this study is representative of a computer program to be written, the estimator who has already determined what his expected man-months will be can also arrive at an estimate of computer hours, using Figure 5; to this extent, Figure 5 can be a useful portrayal of the historical data.

Again, the above development of empirical tools to aid in the estimation of computer programming costs involved nothing new in the way of either principles or techniques; it did involve an analysis of the components of the computer programming process, and the application of available methodology to the study of this process. And as is common in empirical cost research in other fields, the most significant limitation of these studies centered around the collection of adequate cost data. It was this recognition of the inadequacies of ex-post-facto data used in all of these studies, as well as the possibilities offered for more direct management control of computer programming projects, that shifted efforts at SDC toward the development of a cost collection and

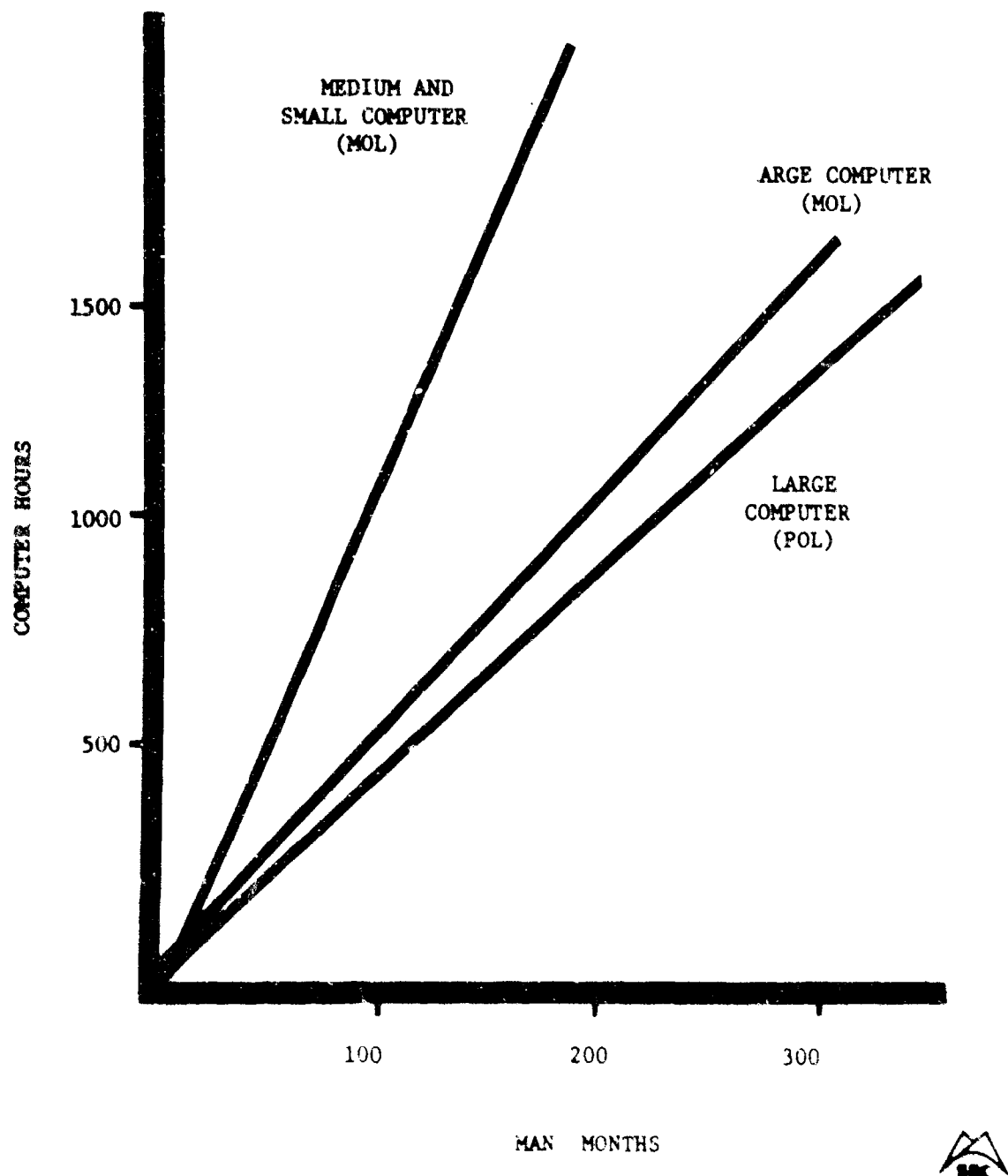


Figure 5. Estimating Computer Hours from Man Months

16 April 1968

17

SP-3122

control system. If resources could be measured as they were expended, a more accurate cost history of computer programming projects could be compiled for use in future research.

A COMPUTER PROGRAMMING PROJECT REPORTING SYSTEM

The Control Process

The managerial function of control involves the measurement and correction of the performance of subordinates in order to make certain that enterprise objectives and the plans devised to attain them are accomplished (1). Control, therefore, implies the existence of plans, and a fundamental principle of control applicable at any level of the organization is that controls can be no more effective than the degree to which they reflect the character, structure, and degree of detail inherent in these plans. Thus, if we are going to control any coordinated human effort, including computer programming, we must be willing to invest the effort required to produce an adequate plan. Perhaps the most significant contribution to management made by the network techniques such as PERT and CPM lies not in their capabilities of reporting deviations or "critical paths" expeditiously, although these are valuable attributes, but rather in the fact that they force managers to plan. To use PERT, tasks and milestones must be defined, interrelationships of these tasks and milestones established, and schedules (and often resources) estimated. These comments suggest that one of the primary reasons why programming projects have slipped schedules and grossly overrun their budgets is that they were not adequately planned in the first place, and hence adequate control simply was not possible.

Programming Project Reporting and Control

In recent years, attention has been given to planning and control of computer programming by commercial computer users (e.g., 14), computer manufacturers (e.g., 15), agencies of the Department of Defense (e.g., 4, 5), and

professional and management organizations (16). The Electronic Systems Division of the USAF Systems Command has sponsored work at the System Development Corporation for the development of a programming project reporting system intended for use by Air Force agencies. There were two objectives of this reporting system:

1. To provide a vehicle for the planning and control of USAF computer programming projects.
2. To collect a data base, from an analysis of which a better understanding of the factors affecting computer programming could emerge. This included the potential for developing more accurate resource estimating relationships.

The resulting product (17) consisted of a system wherein a computer programming project was divided into the nine steps illustrated in Figure 2. These steps were further subdivided into tasks (e.g., Step 8--Information Processing Installation and Implementation--consists of such tasks as file conversion, operational testing, preparation of operating manuals, training, coordination, etc.) for use with larger projects. The end points of these activities (steps and tasks) thus provide milestones for planning. And if all personnel working on a project periodically report the time and resources they spend on each step or tasks, progress can be effectively compared with plans and variances, if any, discovered so that corrective action may be taken.

Figure 6 illustrates a sample summary report that would result from the use of this reporting system. For the sample project shown, only the completion of

PROJECT SUMMARY REPORT

PROJECT - 36608 PROJECT NAME - DAILY EXPENSE AUDIT WEEK ENDING - 67/05/21
 PROJECT LEADER - JONES, L. B. BUDGET, FINANCE AND ACCOUNTING BRANCH IBM 360/40 COMPUTER

ORIGINAL ESTIMATE				ACTUAL / TO DATE		REVISION / EST TO COMPLETE			
S E B	START DATE	COMPLETE MAN COMPUTER	START DATE	COMPLETE MAN COMPUTER	START DATE	COMPLETE MAN COMPUTER	C O M M E N T S		
		HRS MRS RMS		HOURS HRS RMS		HRS MRS RMS			
1	FEASBL. 67/04/10	60	67/04/10	67/04/16	52.0	.	.		
2	ANLIS. 67/04/17	60	67/04/17	67/04/23	55.0	.	.		
3	SYSOSN. 67/04/24	90	67/04/24	67/04/30	87.0	.	.		
4	PRODSN. 67/05/01	180	67/05/01	67/05/17	199.0	.	.		
5	CODCHK. 67/05/15	90	3 12	67/05/15	90.0	4.92 11	67/05/24 14 1 3 1 REVISION		
6	PUNTST. 67/05/22	90	3 12		
7	INTTST. 67/05/29	150	4 12		
8	INSTAL. 67/06/05	120	6 12		
9	MAINTN. 67/06/19	120	4 16		

PROJECT EXPENDITURE VARIANCE (BY COMPARISON WITH ORIGINAL ESTIMATE)

MAN	COMPUTER	COMPUTER
HOURS	HOURS	RUNS

CURRENT VARIANCE - OVERBUDGET / UNDERBUDGET - - 23.0 1.82 1.11

PREDICTED VARIANCE AT COMPLETION - - - - 37.0 2.82 2

Figure 6. A Sample Project Summary Report

the nine basic steps were differentiated as milestones in the original plan. In this illustration, Step 4 has been completed, but the project is behind (with an estimated 3 day slippage for the completion of Step 5), with a current total overrun to date of 23 man hours and 1.82 computer hours; if the overrun continues at the rate experienced to date, a total overrun of 37 man hours is expected.

This project reporting system requires as input data the original estimates by step (and by task if this degree of control is desired), periodic actual expenditures, revised estimates, and comments if any. Outputs such as Figure 6 can then be prepared either manually or by a computer. The value of such a system as an aid in planning and control should be evident from an observation of Figure 6. The value of the system in developing a data base for future research on the economics of the computer programming process is predicated to a large extent on how well comparability between various projects can be maintained; adequate comparability can be achieved if at least the nine suggested steps are used consistently for the planning of all projects, if a standard set of subtasks is consistently used whenever more detailed control is justified, and if some additional descriptive data (e.g., type of application, language used, machine used, etc.) is collected on each project (18).

A. TECHNIQUE FOR EVALUATING SYSTEM EFFECTIVENESS

The previous material in this paper dealt directly with planning, estimating, and controlling the computer programming process. The tools presented are intended for use by operating managers of computer programming projects. By contrast, the following material is far more conceptual and abstract. The purpose is to briefly present the principal elements of a technique for evaluating a total productive system of which operating personnel, computers, and computer programs all play a part. This material is thus intended more for the staff specialist, whose interest is in the creation of management tools and management information systems. Some of the more detailed operational considerations of the application of the proposed technique, with a discussion of the results of a trial using data from the life insurance industry, are discussed elsewhere (2, 19); many of the detailed procedures necessary to adapt the techniques specifically in computer programming management problems still remain to be developed.

The need for criteria and a methodology for evaluating the design and performance of ADP systems has been frequently mentioned in the current literature (20). Of particular interest for this paper is the call for measures in the form of indices (21), since this is specifically achieved with the technique that follows. Index numbers are devices for measuring differences in the magnitude of a group of related variables (22) and are particularly useful for such complex phenomena as the general price level (e.g., the Bureau of Labor Statistics Index of Wholesale Commodity Prices), business activity (e.g., the Federal Reserve Index

of Industrial Production), or qualitative changes or differences (e.g., Storie's Index for Rating the Agricultural Value of Soils).

There has also been a widely expressed (e.g., Mr. Brandon's paper in this session) need for standards of quality, or a means for evaluating the effectiveness of computer programs. The principal difficulty in achieving adequate measures of a computer program's effectiveness, however, is the inescapable fact that computer programs themselves are almost never an end product; rather, computer programs are the means by which computers are used to achieve other purposes. Thus, a measurement that focuses directly on computer program efficiency or effectiveness constitutes, by definition, a sub-optimization. This is why such measures as compile time, throughput time, amount of core used, average process time per run, etc., will never be entirely satisfactory, even if conceptual problems such as the definition of a "typical" job mix or benchmark problem could be resolved.

The technique espoused herein avoids a direct focus on either the computer hardware or the programs by which it operates. Instead, overall measures of the productive process are provided, along with a means for tracing the components of these measures to their origins. The value of computer programs is thus derived by implication from their effect on the performance of the total system of which computer programs and hardware are but a part.

The object of the material to follow is to describe the proposed method for developing evaluation indices for measuring relative overall operating efficiency

of administrative systems augmented by ADP.* This method is applicable to those systems where, for the total system:

1. An objective, numerical measure of the system's output can be devised.
2. Data from a sample of organizations of generally similar outputs can be obtained.
3. More than one input factor (e.g., personnel, and computer hardware) is important to the productive process.

The method borrows from some well recognized tools of the economist, particularly the concept of the production function. The idea behind the production function is that the physical volume of output depends on the quantities of productive agents used in the production process, and the efficiency with which they are used. Although we will direct attention to only two productive agents, labor and capital (or, more specifically, manpower and EDP equipment), it is possible to extend the method to as many productive agents as desired.

* Efficiency is defined here as the attainment of objectives at the least expenditure of resources or, as Harrington Emerson phrased it, efficiency is "...the relation between what is accomplished and what might be accomplished."
(23)

An administrative system is defined as a productive operation whose function consists essentially of processing information or data, and does not involve the physical handling of goods or materials. Administrative systems are a creation of management, a tool to help management do its job of coordination. Computer programming is but one of the many elements that contribute to the development and operation of administrative systems.

When the factors of production may be used in different proportions, that is, when they may be considered as substitutes for one another--the proportions of each productive agent required to produce given outputs may be represented by a curve such as that shown in Figure 7 (and labeled the best-practice production function). This productive-agent curve may be expected to have a shape that is concave to the origin as illustrated.* It represents the technical considerations pertinent to the production process; that is, any point on the curve represents the relative equipment and manpower costs per unit of output that are required to produce that output within existing technological processes. Computer programs are but one part of this process. This curve can be constructed empirically by measuring the outputs of several different systems, and the manpower and equipment inputs that were used to produce these outputs. Each of the ten dots in Figure 6 represents a different system (or organization, or firm) whose outputs are measured in the same units, and whose manpower and ADP equipment inputs are known. The line ABCD connects the points on the concave hull that is closest to the origin. That is, pairs of points are chosen for which the line joining them:

1. Has a negative slope.
2. Is closer to the origin than any observed point.

*Based on the proposition that the greater the quantity of a factor use, the less its marginal productivity will be (¹⁴).

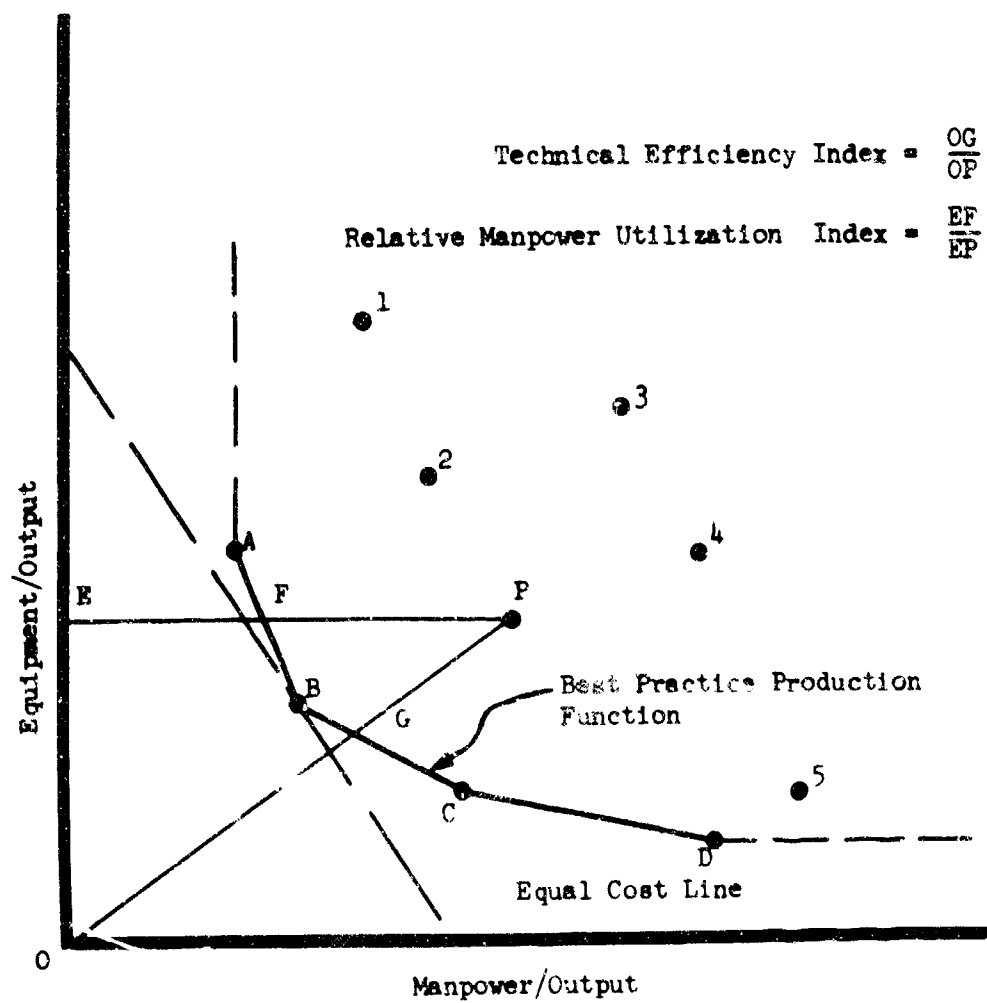


Figure 7. Construction of Indices for Evaluating Over-All Administrative System Performance

To complete the diagram, broken lines are drawn parallel to the axes from the extreme points of the concave hull. Firms A, B, C, and D thus represent the productive performance that is most efficient (minimum use of resources per unit of output) for their own particular mix of resources; theirs are the best combinations of labor and capital attained by any of the firms in the sample. The reasons why one firm exhibits better performance than another (such as the use of more efficient systems, or better computer programs) are not apparent at this point--only the actual differences in performance.

The function constructed in Figure 7 is an approximation of the best-practice production function. The slope of the production function at any point indicates the rate of substitution of labor for capital. The slope AB, for example, indicates the quantity of labor that must be substituted for capital to sustain a constant output when a change is made in the structure of the firm (i.e., in the mix of resources) from that represented by Firm B to that represented by Firm A.

The best-practice production function is a technological relationship, portraying the highest state of the art attained in practice by any member of the sample, for different resource mixes. If an equal-cost line--a line with a slope equal to the unit cost ratios of capital and labor--is drawn tangent to the best-practice production function, the point of tangency represents that firm that also has the lowest-cost combination of resources. This is the optimum firm (Firm B in the example of Figure 7.) Again, this does not mean that Firm B in Figure 7 has the best computer programs; only that its total overall operations,

including its hardware and software resources, are currently producing better results at less cost than the other firms.

The construction of a best-practice production function in the manner described above implies that all firms on the best-practice curve are operating at peak efficiency; if output were increased, the amount of resources employed would therefore have to increase correspondingly. This may not be strictly true, since indivisibility of units of a resource may provide some reserve capacity. Also, a best-practice firm may lead the field to such an extent that its actual performance substantially understates its capability. Such considerations mean that the proposed method for arriving at a best-practice production function is conservative in the sense of producing a realistically obtainable target for an administrative system to attain. That is, investment in system changes can be expected to result in total cost savings up to that measured by the difference between current operations and the best-practice target.

Having constructed a production function representing the best-practice standard for a sample of firms, we are now prepared to build indices relating the performance of any given firm to this standard.

There are many ways of comparing the performance of a given firm with the standard represented by a best-practice production. In a more elaborate description of this method (19), a total of twelve evaluation indices were advanced; at this time I will illustrate the potentials by describing only two: the Technical Efficiency Index, and the Relative Manpower Utilization Index.

To construct the Technical Efficiency Index for the system represented in Figure 7 by point P, point P is connected to the origin by a straight line intersecting the best-practice production function at point G. The technical efficiency of system P is then represented by:

$$\text{Technical Efficiency Index} = \frac{OG}{OP}$$

This index, a contribution of the English economist, M. J. Farrell (25), measures the success of system P, relative to a hypothetical firm G (firm G consists of a weighted average of the most appropriate observed firms on the best-practice production function), in producing maximum output from a given set of inputs. Technical efficiency, as here defined, compares systems on the basis of equivalent mix of inputs. Much of the difficulty in making intersystem comparisons, as between systems P and S in Figure 7, arises from the objection that such systems are so fundamentally different that a comparison would be meaningless. We have previously assumed roughly equivalent outputs; the technical efficiency index attempts to achieve comparability on the input side, substantially mitigating this objection.

Since technical efficiency is a function of the best-practice production function, the evaluation of a firm depends not only on that firm's realized achievement, but on the possible achievement available to it within the constraints of technology. Even if a firm improves its performance, its technical efficiency index will decrease if even greater improvement is achieved by other firms (this would be represented by a greater movement toward the origin by the production function in Figure 7 than by the point

in question). Another important property of technical efficiency is that the lowest-cost firm in the sample will always have perfect technical efficiency, but so may a number of other firms operating with different proportions of inputs.

The Relative Manpower Utilization Index is formed (Figure 7) by drawing a straight line from point P parallel to the manpower expense axis, intersecting the production function at F and the equipment-expense axis at E. The Relative Manpower Utilization Index of system P is then represented by:

$$\text{Relative Manpower Utilization Index} = \frac{EF}{EP}$$

This index is a measure of the degree to which manpower expense can be reduced (within the limits of the technology) assuming that equipment expense remains constant. The index may find use in the evaluation of current manual procedures and various personnel factors. It is superior to simply comparing labor productivity to that of the least-total-cost firm (point B, Figure 7) or the firm with the lowest manpower expense per unit output (point A, Figure 7), because it explicitly considers the contribution of the equipment resource (and the computer programs that are included with it) available at the pertinent manpower/output level.

Both the Technical Efficiency Index and the Relative Manpower Utilization Index are ratio quantities. And there is always some question as to whether ratio measures can be conceptually adequate measures of system performance. Since this is an important consideration to the usefulness of the technique

just described, a discussion of the appropriateness of ratios as measures of effectiveness is in order.

The reason often advanced for avoiding ratios in cost-effectiveness studies is that optimal ratios like optimal computer programs, per se, are usually not the primary goal. A distraction from primary objectives can lead to ridiculous conclusions. For example, the selection of a house on the basis of the least cost per square foot could result in the choice of a \$500,000 mansion rather than the \$25,000 bungalow that may more closely meet the real needs and budget of the purchaser.

Because absolute magnitudes are important, the general statement of those criteria appropriate for cost-effectiveness studies distinguishes the following (19):

1. Fixed gain, variable cost. Resources are added to the various alternatives up to the point where each alternative accomplishes the objective; the best alternative is that with the least absolute cost.
2. Fixed cost, variable gain. The alternative is chosen that accomplishes the most objectives (or the greatest degree of a single objective) at a given cost.
3. Maximize absolute difference between gain and cost.

Each of these three criteria is generally acceptable; each has its advantages, depending on the problems of measurement and the circumstances of a particular problem. The fixed-gain case is especially applicable to problems in which achievement of the objective is binary--i.e., you either win or lose (this would be appropriate for many military decisions). Many situations in which

gain is difficult to measure may be handled with the fixed-gain ("just meet the specifications") criterion. The fixed-cost case is, of course, most useful when the major constraint on the problem is a fixed budget; it can also be a convenient criterion when there are many subsidiary objectives. The fixed-cost and the fixed-gain criteria are equivalent, if the size of either gain or cost is the same in both of the two tests; that is, if the best alternative for a \$100 budget produces a gain of 50, then the least-cost choice for a fixed-gain of 50 would be the same alternative, which costs \$100. Therefore, the choice between the fixed-cost or the fixed-gain criteria depends largely on whether cost or gain can be more readily fixed in the particular analysis in question.

The application of the third criterion--maximizing the difference between gain and cost--depends on the ability to measure gains and costs in the same kinds of units. This criterion is the same as the familiar business criterion of "profit maximization." Stated in the economist's terms of total unit costs and outputs, the optimum (maximum gains) occurs at the output level at which marginal costs equal marginal revenue. When opportunity costs are considered--that is, when the gains forfeited by not choosing an available alternative are included in the costs of the remaining alternatives--to maximize gains minus cost is the same as maximizing total gains.

The reason for the above elaboration is that under certain circumstances, a ratio may in fact be simply a restatement of the three generally acceptable criteria. The contention is that for most business firms, profit maximization

is still the best assumption as to the objective of the firm, and that internal efficiency is the handmaiden of profit maximization. If long-run unit costs do not necessarily increase with increased output (i.e., if there are no significant diseconomies of large-scale production), and there is good reason to believe that this is the case (2), there is no incentive for the firm to restrict output; on the contrary, the firm would tend to increase its output to the limits of the market. Under these conditions, with the objective of making absolute size as large as possible, the ratio of costs to the outputs becomes an excellent measure of efficiency; it is equivalent to either the minimum cost at fixed gain or the maximum gain minus cost criteria. The best ratio indicates the preferred system, no matter what the scale.

One of the problems in constructing indices in the manner suggested herein is that it is not always easy to find situations where the three basic requisites of the method (objective measure of output; sample of systems with generally similar outputs; more than one important input factor) are met. However, there are many cases where these methods do apply. The potential for a 50 percent saving in manpower expense in the life insurance industry cited earlier in this paper was obtained by the use of the Relative Manpower Utilization Index. And frequently, minor variations in the operations of different systems within a sample can be accounted for with such standard techniques as multiple regression analysis. These matters are dealt with at length elsewhere (2, 19).

Before leaving the topic of evaluation indices, one further point is in order. This relates to the meaning of such overall productivity indices as defined herein to the management of the computer programming process. The meaning for programming management is simply that the productivity of the system, as measured by the Indices, is a reflection of the ultimate success of the total system; computer programs are merely one element in this productive process. Even if the measurement of the quality or efficiency of computer programs poses difficult conceptual and empirical problems, ultimate system productivity can be quite objective; and computer programs have little value if they do not have an impact on the ultimate objective of the system. Thus we return again to basic management principles, in this case management by objectives (26), to develop a tool for system evaluation. It is not necessary, however, to look only at the final productivity figures in using indices such as those suggested here; on the input side, the components of these indices, can be traced back to their source, as illustrated by Figure 8. Thus, a series of related indices can be constructed; such a series should be quite useful for exploring the causes of variations in total systems productivity.

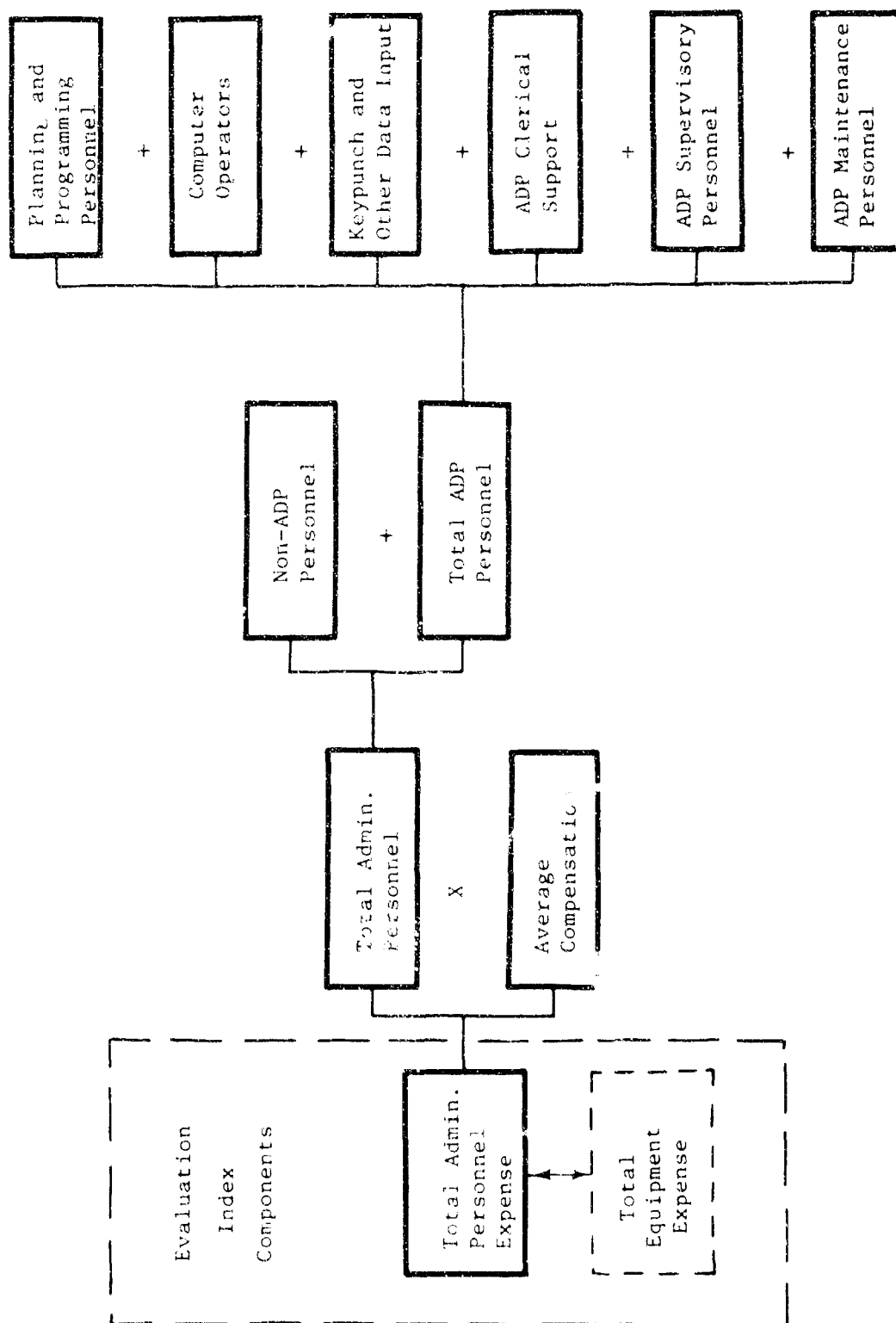


Figure 8. One Possible Breakdown of Total Administrative Personnel Expense

SUMMARY AND CONCLUSIONS

I have discussed several useful tools for the computer programming manager developed at the System Development Corporation: a planning guide; empirically derived computer program cost estimating guides; a project reporting and a data collection system. I have also described a few of the basic concepts of a method for evaluating the comparative efficiency of EDP augmented administrative systems with measurable outputs. None of these tools are revolutionary or novel in their design or application; on the contrary, their development illustrates the application of well established management principles.

Both the planning guide and the cost estimating handbook have been widely distributed to government agencies and also to a number of private corporations. I have not made any direct effort to measure the benefits to these organizations from the use of these materials, but the comments received to date indicate a gratifying acceptance. The project reporting system has been delivered to the Air Force Data Systems Design Center for their use. And, as mentioned earlier, the productivity evaluation indices have been successfully applied to test several management hypotheses for a sample of firms in the life insurance industry.

The experience received from working on the programming management tools described above results in the following general observations:

1. Cost data should be collected as the programming project proceeds, not after the fact, if cost prediction with accuracy greater than that

demonstrated by the material presented in this paper is desired.

2. To develop useful estimating relationships with acceptable predictive power, the analysis should have a comparatively narrow focus, such as on specific languages and/or applications. For example, a study of COBOL programming for inventory applications could be of considerable value, both as a research vehicle for measuring the impact of important factors (e.g., programmer experience) and for developing accurate and dependable estimating relationships.
3. All permanent programming organizations should collect cost data on their own operations. This would enable the development of estimating relationships that are directly pertinent to each organization's own mix of resources, products, and particular environment, as well as promote better project control. The extent and detail of the data collection would depend upon the particular operations; however, the determination of at least the total resources expended per project is recommended for all operations.
4. The basic structure for planning, presenting data (the handbooks), or collecting information (the reporting system), developed by SDC could be used even without modification by all computer programming organizations. Some adaptation of this material, however, would probably be advisable (e.g., the level of detail at which costs are collected would depend upon the size of programming projects). The numerical cost estimation material from SDC's research (e.g., Table I and Figures 3, 4, and 5) should be used with extreme caution, not only because of the validity of data collected ex-post-factor by questionnaire, but because

the sample studied by SDC may not be representative of the program mix of the user.

5. Evaluation indices of the type described herein are empirically workable, and are conceptually adequate measures of system performance for certain kinds of systems. They are recommended for use in research on the behavior of organizations and the ultimate impact of computer programs. Also, the construction of these indices and the tracing of their components to their sources could be used as a foundation for a management information system.

The major conclusion, however, is the basic proposition with which this paper was introduced: that the most expeditious means to enhance the management of computer programming today is to apply to the computer programming process the principle of management currently successful in other coordinated activities.

References

1. Koontz, Harold and Cyril O'Donnell, Principles of Management, McGraw-Hill Book Company, New York, Third Edition, 1964.
2. Nelson, Edward A., Firm Size and Administrative Efficiency in the Life Insurance Industry. Ph.D Dissertation, UCLA, June 1968.
University Microfilms, Inc., Ann Arbor, Michigan.
3. LaBolle, Victor, L. Farr, N. E. Willmorth, Planning Guide for Computer Program Development, TM-2314/000/00, System Development Corporation, Santa Monica, California, 10 May 1965 (AD 465-228).
4. Ratynski, Milton V., "The Air Force Computer Program Acquisition Concept," AFIPS Conference Proceedings, Vol. 30, Spring Joint Computer Conference, 1967, pp. 33-44.
5. Searle, Lloyd V., and George Neil, "Configuration Management of Computer Programs by the Air Force: Principles and Documentation," AFIPS Conference Proceedings, Vol. 30, Spring Joint Computer Conference, 1967, pp. 45-49.
6. Piligian, M. S. and J. L. Pokorney, "Air Force Concepts for the Technical Control and Design Verification of Computer Programs," AFIPS Conference Proceedings, Vol. 30, Spring Joint Computer Conference, 1967, pp. 61-66.

7. Searle, Lloyd V., and R. L. Henderson, System Engineering Guide for Computer Programs, TM-3596, System Development Corporation, Santa Monica, California, 15 August 1967.
8. _____, Systems Management Aspects of Computer Program Test and Activation, TM-3361/000/01, System Development Corporation, Santa Monica, California, 15 August 1967.
9. Searle, Lloyd V., and George Neil, Configuration Management of Computer Programs for Information Systems, TM-1918/000/01, System Development Corporation, Santa Monica, California, 12 July 1965.
10. LaBolle, Victor, "Development of Aids for Managers of Computer Programming," Journal of Industrial Engineering, November, 1966.
11. Nelson, Edward A., Management Handbook for the Estimation of Computer Programming Costs, TM-3225/000/01, System Development Corporation, Santa Monica, California, 20 March 1967 (AD 648-750).
12. Gradwohl, Alan J., George S. Beckwith, Stanton H. Wong, Phase II Final Report on Use of Air Force ADP Experience to Assist Air Force ADP Management, ESD-TR-66-671, Vol. II, Planning Research Corporation, Los Angeles, California, December 1966 (AD 646-870).
13. Weinwurm, George F., On the Economic Analysis of Computer Programming, SP-2772, System Development Corporation, Santa Monica, California, August 1967.

14. Anderson, Robert M., "Management Controls for Effective and Profitable Use of EDP Resources," Proceedings of 21st National Conference - ACM, Thompson Book Company, Washington, D. C., 1966.
15. Newhall, H., A Report on IPERT (Installation Programming Evaluation and Review Technique), and informal report of the Honeywell EDP Division, Los Angeles, California, 1963.
16. Lecht, Charles P., The Management of Computer Programming Projects, American Management Association, Inc., 1967.
17. Petersen, Kenneth E., R. N. Clark, E.-A. Nelson, Data Automation Resource-Reporting and Management System, TM-3558/000/01, System Development Corporation, Santa Monica, California, 11 August 1967.
18. Nelson, Edward A., T. Fleishman, A System for Collecting and Reporting Costs in Computer Program Development, TM-3411/000/00, System Development Corporation, Santa Monica, California, 11 April 1967 (AD 657-793).
19. Nelson, Edward A., The VOPIR Technique for the Evaluation of Computer Augmented Administrative Systems, SP-2765, System Development Corporation, Santa Monica, California, 17 May 1967.
20. For example, see: Bureau of the Budget, Report to the President on the Management of Automatic Data Processing in the Federal Government (the Clewlow Report), U. S. Government Printing Office, 4 March 1965, p. 18.

16 April 1968

42
(last page)

SP-3122

21. Committee on Post Office and Civil Service, House of Representatives, Use of Electronic Data Processing Equipment in the Federal Government, U. S. Government Printing Office, 16 October 1963, p. 7.
22. For a general description of a number of indices, see: Croxton, F. E., and D. J. Cowden, Applied General Statistics, Prentice-Hall, Inc., Englewood Cliffs, New Jersey; Second Edition, 1955, pp. 436-448.
23. Drury, H. B., Scientific Management, New York, Columbia University Press, 1915, pp. 114-115.
24. McKenna, Joseph P., Intermediate Economic Theory, Holt, Rinehart and Winston, Inc., 1958.
25. Farrell, M. J., "The Measurement of Productive Efficiency," Journal of the Royal Statistical Society, Series A (General), Vol. 120, 1957, pp. 253-290. The Technical Efficiency Index is also similar to Debreu's "Coefficient of Resource Utilization," in Debreu, G., Econometrica, Vol. 19, 1951, pp 273-292.
26. Koontz, p. 432.

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) System Development Corporation Santa Monica, California		2a. REPORT SECURITY CLASSIFICATION Unclassified	
3. REPORT TITLE Some Recent Contributions to Computer Programming Management.		2b. GROUP	
4. DESCRIPTIVE NOTES (Type of report and inclusive dates)			
5. AUTHOR(S) (First name, middle initial, last name) Edward A. Nelson			
6. REPORT DATE 16 April 1968		7a. TOTAL NO. OF PAGES 43	7b. NO. OF REFS 26
8a. CONTRACT OR GRANT NO.		9a. ORIGINATOR'S REPORT NUMBER(S) SP-3122	
b. PROJECT NO.		9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
c.			
d.			
10. DISTRIBUTION STATEMENT Distribution of this document is unlimited.			
11. SUPPLEMENTARY NOTES		12. SPONSORING MILITARY ACTIVITY	
13. ABSTRACT <p>The System Development Corporation (SDC) has been conducting research in the management of the computer programming process since 1963. This paper briefly describes some of the results of this research, plus individual research by the author, and the relevance and usefulness of the results to the operating manager.</p> <p>The paper emphasizes four specific types of management tools that were produced: (1) planning aids; (2) cost estimating guides; (3) a project reporting and control system; (4) a technique for evaluating the effectiveness of certain classes of computer-centered information systems. Each of these tools is briefly described, and the research design and procedures used in their development are mentioned. The experience gained -- at SDC and elsewhere -- in applying traditional research techniques to the computer programming process has yielded certain insights regarding the economic and management dimensions of computer programming, and several of these insights are discussed. Foremost among this author's conclusions is that the management principles that apply to any other coordinative activity are equally applicable to the computer programming process. Specific techniques may differ, especially at the lower echelons, but these differences pertain to the technical skills and procedures of the production process, they are engineering, not management or economic issues. The paper concludes with several general observations pertinent to future research in computer programming economics and management, and the use of the management tools described.</p>			

Unclassified

Security Classification

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Computer programming Program management Planning aids Cost estimation Information systems						

Unclassified

Security Classification