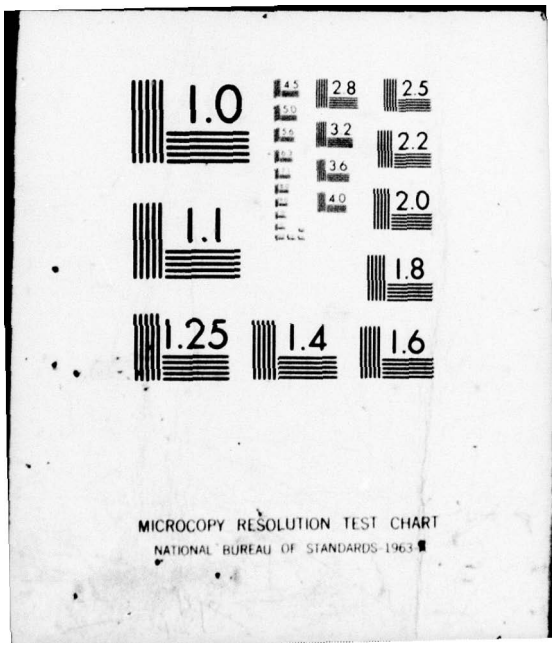
AD-A036 066 SYSTEM DEVELOPMENT CORP SANTA MONICA CALIF F/G 9/2
SOFTWARE DATA COLLECTION STUDY. VOLUME V. SURVEY OF PROJECT MAN--ETC(U)
DEC 76 N E WILLMORTH F30602-75-C-0248
UNCLASSIFIED SDC-TM-5542/005/01 RADC-TR-76-329-VOL-5 NL

1 OF 1
AD A036066

END

DATE FILMED
3-77


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963

RADC-TR-76-329, Volume V (of eight)
Final Technical Report
December 1976

SOFTWARE DATA COLLECTION STUDY
Survey of Project Managers

System Development Corporation

This report has been reviewed by the RADC Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public including foreign nations.

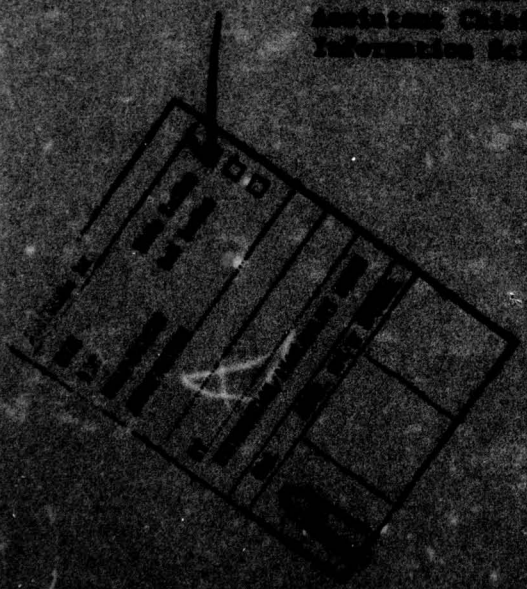
This report has been reviewed and is approved for publication.

APPROVED: Richard T. Slawinski

RICHARD T. SLAWINSKI
Project Engineer

APPROVED: Alan R. Barnum

ALAN R. BARNUM
Assistant Chief
Information Sciences Division



FOR THE COMMANDER: John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

REPORT NUMBER: RADC-TR-76-329-Vol [redacted]-5	2. GOVT ACCESSION NO. / 3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): SOFTWARE DATA COLLECTION STUDY. Volume V. Survey of Project Managers.	5. TYPE OF REPORT & PERIOD COVERED: Final Technical Report. June 1975 - June 1976.
	6. PERFORMING ORG. REPORT NUMBER: TM-5542/005/01
7. AUTHOR(s): N. E. Willmorth	8. CONTRACT OR GRANT NUMBER(s): F30602-75-C-0248
9. PERFORMING ORGANIZATION NAME AND ADDRESS: System Development Corporation, 2500 Colorado Avenue, Santa Monica CA 90406	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 63728F, 55580810
11. CONTROLLING OFFICE NAME AND ADDRESS: Rome Air Development Center (ISIS), Griffiss AFB NY 13441	12. REPORT DATE: December 1976
	13. NUMBER OF PAGES: 35
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office): Same	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer:
Richard T. Slavinski (ISIS)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Software Data
Data Requirements
Collection Problems

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Due to the paucity of data in the technical literature dealing with the problems, procedures and data requirements for the management and scientific investigation of software development, a survey was conducted sampling SDC project managers and military program offices. The survey asked questions concerning the willingness to supply data, sources of unreliability, the classes of data supplied and needed, and current data collection practices.

Responses were received from all three services as well as SDC managers

OVER

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

339900 AB

(about two-thirds of the military responses were from Air Force Agencies.) Project managers were relatively willing to release all classes of data except direct dollar costs, although feelings varied widely. All projects had an organized method of planning and producing software, but few thought actual planning was adequate. (In general, projects felt that they were rushed into code production without adequate designs.) Most projects endorsed the utility of reviews, but less than half felt the reviews were adequate. Most projects did not use military standard practices, but had internal procedures. Procedures were critized for lack of standardization, and inadequate information. Managers controlled their projects through progress reports rather than more detailed developmental information, and felt the information they got were most inadequate for the less well-structured areas of program development: analysis and design. The data they received was rated non-standard, subjective, and of questionable accuracy and validity.

Automation, standardization, systemization with full understanding of needs and provisions, and independent verification through audit checks were seen as the most promising solutions to data collection problems.

TABLE OF CONTENTS

SOFTWARE DEVELOPMENT DATA COLLECTION:

SURVEY OF PROJECT MANAGERS

1. INTRODUCTION

The objective of this survey was to gain an assessment from first hand experience regarding the problems associated with the collection of software development data. When the study of software data collection problems was first undertaken (See Volume 002 of this report), the project members intuitively expected the literature to provide such an assessment, an intuition that proved unfounded. There were actually very few studies reporting real experience in the collection of data and not many more speculating about them. To fill this lack, the project prepared a questionnaire covering the problems of data collection as these were tentatively revealed by our initial investigations and administered it to a sample of project managers at SDC and at Program Management Offices in the military.

Responses were received from 15 SDC project managers stationed at Santa Monica, Colorado Springs, Washington, and Huntsville. Responses were received from 10 (largely with sparse answers) Military program management personnel, two thirds from Air Force Agencies (ESD, AMC, SAMSO, SAMTECH) and the remainder from Army and Navy offices. Response rates were about 30% from internal SDC sources and 20% from military agencies.

2. PROJECT CHARACTERISTICS

Most projects were engaged in scientific, engineering or R&D projects with very few in business data processing, as might be reported. Time sharing and batch were about equally employed with a great many projects using both and a few using remote batch (4 projects out of 25). Almost all SDC projects reporting were engaged in federally sponsored developments, but some were dealing with private industry, others with local government and some were

internally sponsored. The computers used were very heavily IBM 360/370 or HIS 6000 series machines, with a sprinkling of CDC, Xerox, UNIVAC, and Burroughs machines, plus some minicomputers in communications and avionics applications. The only specialized peripherals widely interfaced with were communication equipment, but four projects dealt with avionics and sensor equipment and two with navigation gear.

Project size varied from 2 persons and 5000 object instructions to 200 persons and 1.2 million instructions. Modularization used on the projects did not seem extreme, ranging from around 300 to 2000 object instructions per module or routine. Languages used include FORTRAN, JOVIAL, COBOL, GMAP, BASIC and assembly language. Experience ranged from 2.5 to 15 years with a modal value of 8. Projects ranged from 8 months to 6 years in length.

Almost all projects used compilers, utilities, dumps, link editors and program libraries. Nearly as many used debug packages, recording/reduction tools, and flow charters. Almost a third of them used simulators and data management tools and a sprinkling used verifiers, auditors and timers. Half the projects thought their tool package quite adequate, some (10%) thought they were marginal and a few (15%) thought them insufficient. Additional tools suggested included project management and scheduling, configuration management, automatic documenters, and automatic verifiers for all languages. Graphic output and debugging packages that are easier to use, less constraining and having few instrumentation effects (i.e., whose use forces particular structure on the tested item) were suggested. Tools were also criticized for being inadequately standardized--compilers too diverse in the efficiency of code produced and the errors checked, data management programs were too specialized, flow charters existed at several levels of detail, and compliance with specification standards was judged very variable.

Except for the R&D projects, projects are of modurate size, produced by experienced craftsmen using traditional or standard tools and techniques. The possible exception lies in the large proportion of projects using inter-

active programming at least part of the time.

3. DATA SENSITIVITY

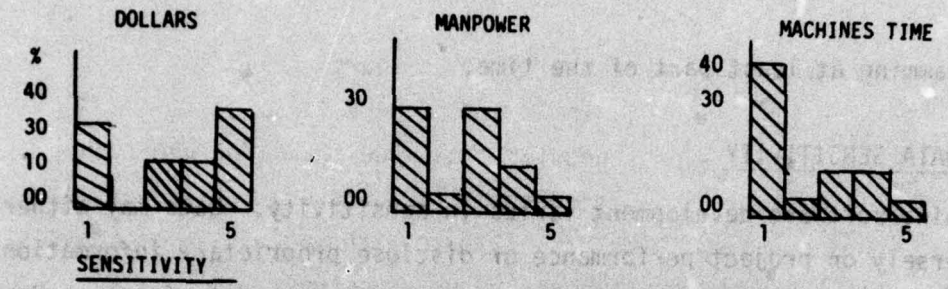
Data concerning software development varies in sensitivity. Data may either reflect adversely on project performance or disclose proprietary information to competitors. Managers may be reluctant to release such data freely. However, to be maximally useful for comparative studies of software methodology and reliability, all types of project data are desirable. In this questionnaire the respondents were asked to rate their relative reluctance or willingness to release data to a semipublic data bank given reasonable guarantees of privacy to avoid open criticism of project performance.

Unfortunately, quite a few of the military PMO personnel did not respond to this section on the apparent grounds that such reporting did not apply to their operation. It was intended that the PMO's rate their contractors in terms of the resistance the PMO encountered in gathering data from them, but either the questionnaire instructions failed to make this clear or the project management offices did not collect such data or experience customer behavior in regard to it. Consequently, no separate analysis for PMO responses is reasonable; the ratings of those who did respond are included in the internal counts shown in the following figures.

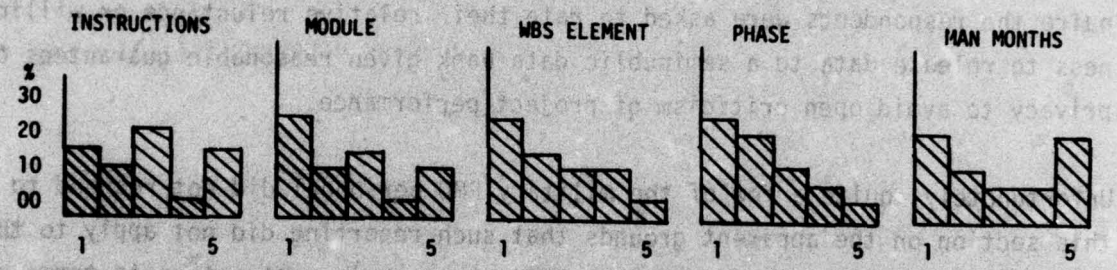
There was a wide range of opinion on every item. On every item at least someone said that they would be very reluctant and someone else said they would be very willing. (A rating of '1' indicates willingness, a rating of '5' indicates reluctance in the figures.)

Figure 1 shows cost and schedule data ratings. To summarize, the project managers were most reluctant to release dollar costs, rather willing to release costs in manpower and very willing to release costs in machine time. Although not pronounced, there is some greater reluctance to release fine costs (costs

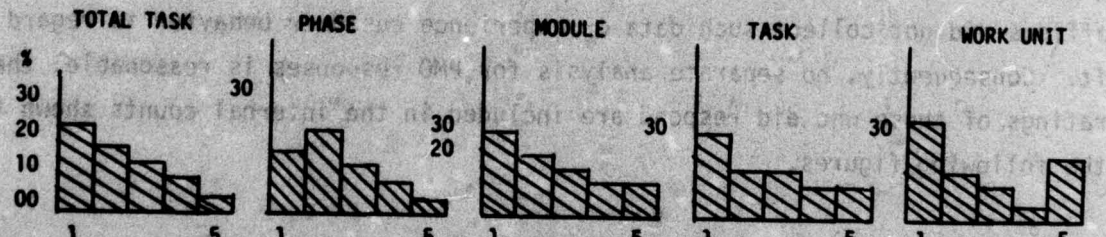
1. COST DATA



2. RELATIVE COSTS PER:



3. SCHEDULE BREAKDOWN LEVEL



4. COST VARIANCE

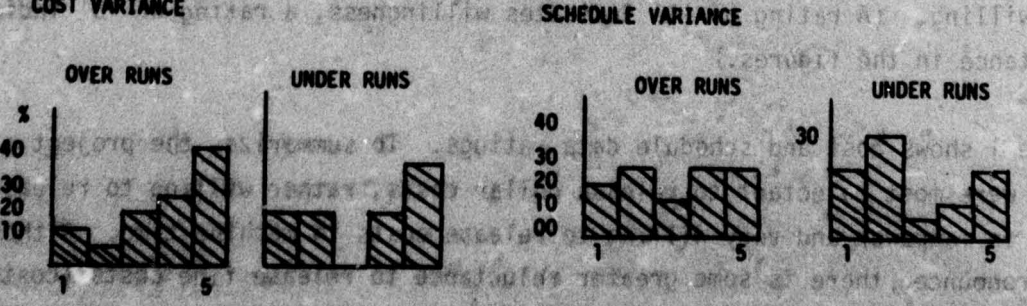


Figure 1. Data Sensitivity

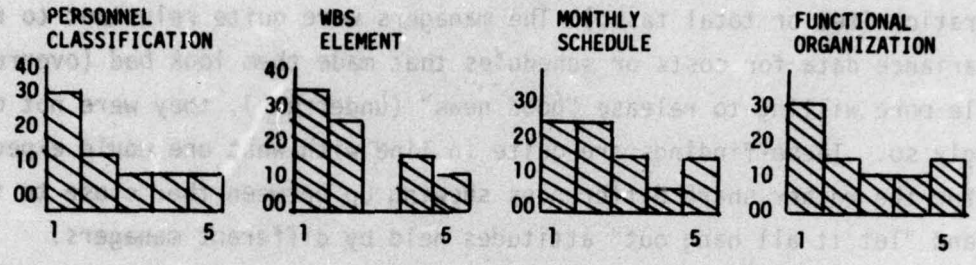
per instruction or per individual activity) than coarse costs (costs per total configuration item or total task.) The managers were quite reluctant to release variance data for costs or schedules that made them look bad (overruns) and while more willing to release "good news" (underruns), they were not overwhelmingly so. These findings are quite in line with what one would expect, except for the rather sharp differences showing up between the "close to the vest" and "let it all hang out" attitudes held by different managers.

Figure 2 shows the relative willingness or reluctance of managers to release resource utilization information. Various manpower breakdowns received moderate indorsement (more favorable than unfavorable) and computer time breakdowns was very definitely approved. (There were no facility managers in the sample; it might have made a difference.) Surprisingly, the managers were quite willing to release personnel turnover data whether that data were for project members or for key technical and managerial personnel.

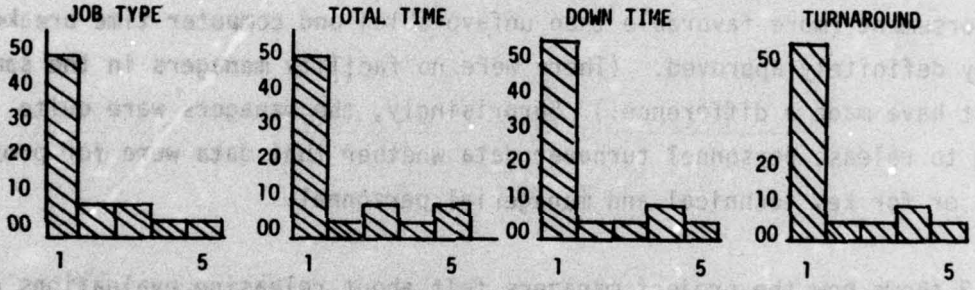
Figure 3 shows how the project managers felt about releasing evaluations of project performance whether these were proficiency ratings of personnel or computing facility efficiency. Project managers were quite ambivalent about releasing personnel proficiency ratings--as can be seen they scattered their ratings fairly uninformally across the spectrum--but quite willing to release experience figures. Again, for productivity ratings they were somewhat restrained about releasing work unit costs or durations, but were reasonably willing to rate the computer and other support activities on their efficiency. Again, except for the relative willingness of some managers to release proficiency ratings of their project members, those findings are fairly well in line with what one would expect.

Figure 4 is also fairly well in line with what one would expect--managers are quite willing to release configuration data--unless evaluative ratings of efficiency, reliability, etc., are involved. Even here the managers are more willing than not to share their experiences.

1. MANPOWER BY



2. COMPUTER TIME BY



3. PERSONNEL TURNOVER DATA BY

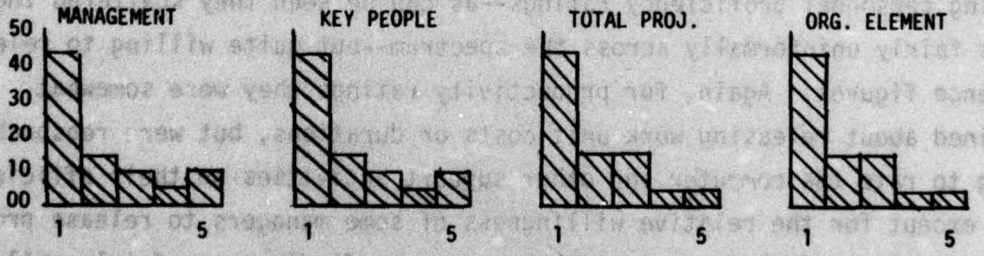


Figure 2. Resource Utilization Data.

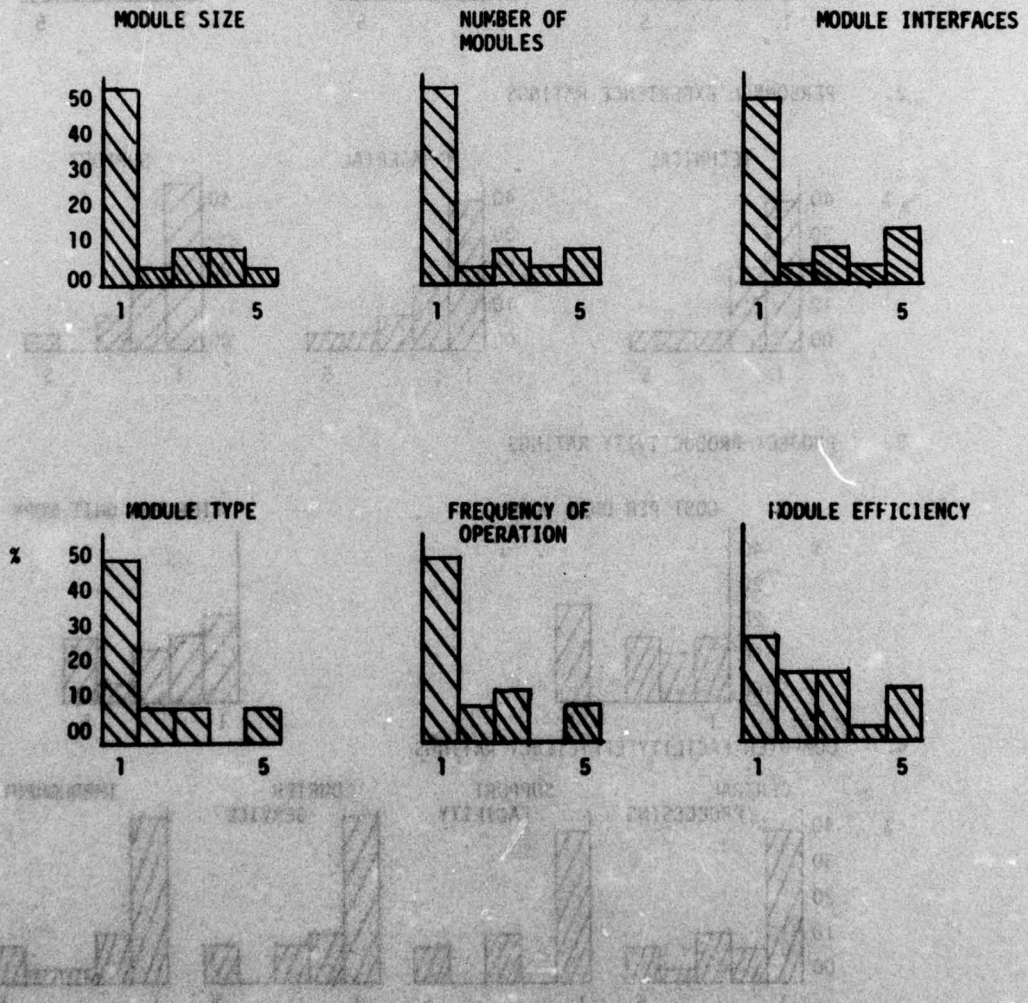
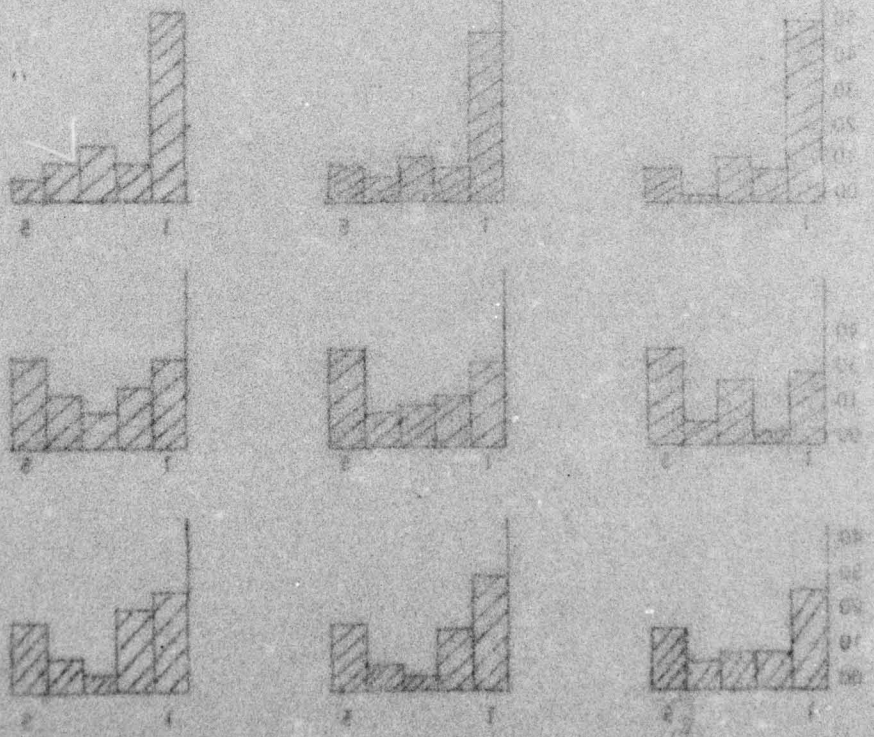


Figure 4. Module Statistics.

Figure 5 compares the sensitivity of data concerning various events that might lead to program changes: requests for modification, reports of problems encountered, and reports of suspected program errors. The results are quite in line with previous findings. Managers are quite willing to release objective information about the numbers, sizes and types of changes encountered, but more reluctant to release data on cost and schedule impacts. Again, the division between the willing and the reluctant is quite plainly polarized. No information is readily available from those two parties (the returns were all anonymous, effectively prohibiting follow up). However, a more penetrating inquiry might yield interesting results.



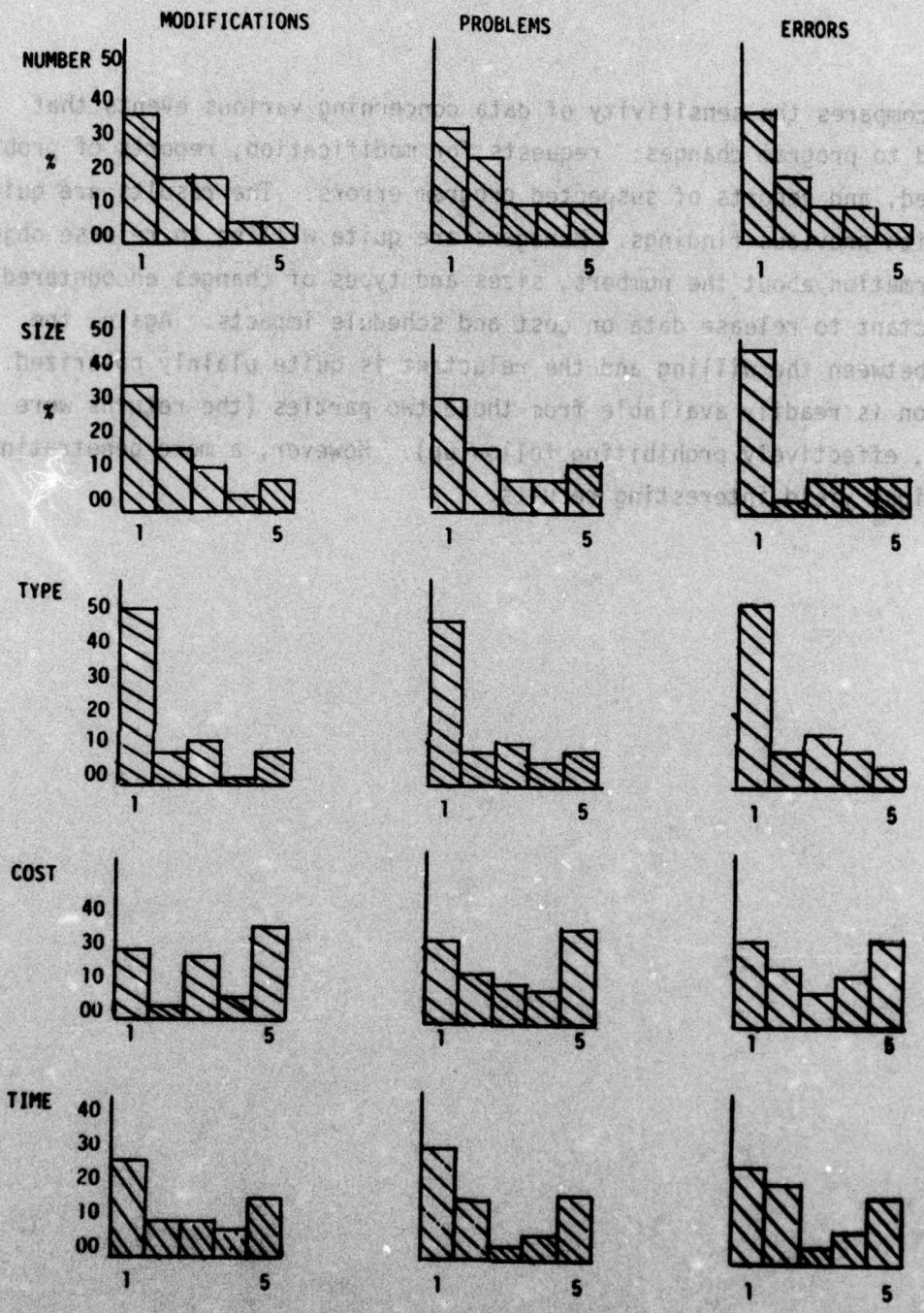


Figure 5. Modification Statistics.

4. SOFTWARE DEVELOPMENT PROCESS

Software is developed and managed in a variety of ways. Currently, new technologies are being tried and the most cost effective and reliable methods have not been determined. However, it seems a number of project managers have solved many of the problems of software development to their own satisfaction. Their answers to several questions aimed at soliciting satisfaction with the suggested improvements to their current methodology can be taken as the criterion.

4.1 PLANNING

Almost all respondents stated that they had an organized method for producing software for all phases of development. Similarly, plans were laid for manpower utilization, schedules and budgets for work, project organization, financial matters and testing. Those with documentation plans and configuration management plans were only slightly fewer. More specialized plans, like those for facilities, training, conversion, support and liaison fell to around 25% of the respondents. However, when asked to evaluate the adequacy of planning, only six respondents felt planning was adequate. Ten said outright that it was not and four said somewhat or only in special projects. Comments received on this item tended toward criticizing product plans (i.e., designing) as being inadequately done. Some of the comments include:

> "Monetary constraints tend to drive projects into production phase prior to proper design completion--customer is to blame."
>
> "First major delivery is almost always too early to allow adequate system design and project planning."
>
> "Usually [adequate], although the time required for these activities [development and test] and product reliability can both be improved by allotting more time to design."

"[Planning could be improved by] a detailed work breakdown structure."

4.2 KEY PERSONNEL

Approximately 40% of the respondents said at least some personnel were specified by the proposal and/or the contract. Thus current practices do reflect an emphasis upon key technical resources.

4.3 INDEPENDENT TEST TEAMS

One of the proposed techniques for ensuring greater reliability of software is the utilization of independent test teams to perform final design verification and validation. Less than a third of the project managers surveyed said they used such teams. When used, such test teams were seen as doing an effective job from the customers point of view. The few comments made regarding these questions said:

"Monitary constraints prohibited this concept [on this project]. Consequently, many problems occurred in developing adequate, for the customer, test plans and procedures."

"Independent tests [and reviews] work best when the schedule is allowed to slip to accommodate delays in the review process."

"[Independent test teams were] planned, but not effectively applied."

"Partially [adequate] from the customer's point of view, although a great part of their effectiveness derives from the adequacy of the Part I specs and the functional breakout of the CPCI [for the test team to work from]."

4.4 SOFTWARE REVIEW PROCESSES

The use of software reviews to increase the reliability of software, especially during the critical early period of software development, has received considerable attention in recent years. Respondents were asked if they did have a set of consistent and periodic reviews and, if so, which of the standard military reviews had been instituted. Two thirds of the respondents said they did use systematic review procedures. Even more (nearly 80%) used Preliminary Design Reviews (PDR's) and Critical Design Reviews (CDR's). Only 50% however, indicated they used Performance Requirement Reviews (PRR's) or Formal Qualification Reviews (FQR's). To this list, however, the military respondents added System Requirements Review, (SRR) and Functional and Physical Configuration Audits (FCA and PCA), and the project managers added informal design reviews, structured walk throughs, and technical interchange meetings.

The respondents were asked whether or not there was a systematic procedure for incorporating the discrepancies found in the reviews into the software development product in a timely and cost effective way. They were also asked if the customer was involved in the review and discrepancy resolution process and, if not, would it help if they were. About 40% of the respondents said they had such procedures, but the comments to the question indicated that the procedures varied from formal design change and error report processing to informal day to day interactions. About the same proportion, with some qualifications, indicated that the customer was involved in software reviews and that his attendance was helpful. Some comments received were:

> "The project had excellent customer involvement; all changes were designed, costed and scheduled practically on a day to day basis. Customer personnel were on the technical team [a joint programming project] and participation was excellent."

"[Discrepancies were resolved] by use of Design Modification Requests and Discrepancy and Correction Reports and at the larger, more formal reviews by issuing Action Items. More customer attendance at reviews would be desirable."

"Depends upon what you call "systematic". Discrepancies are certainly corrected. More customer participation would probably hinder the operation. Our customers are mainly concerned with the final product and not generally concerned with intermediate reviews."

"Not sure [if more participation would help]. May complicate things."

"Our customer sits in judgment at all design reviews and has a seat on internal review committees. However, he typically does not understand the system and consequently slows down the progress made."

"After a baseline, standard ECP processes are used; Design Modification Requests to the Design Control Group at any time. Customer personnel are in house on the project. Their participation in design reviews is absolutely imperative to avoid argument and delay."

"Modifications are made to the design document prior to formal implementation of the required changes. Customers attend informal as well as formal reviews."

"Customer participates, but is usually unqualified to contribute effectively. If the customer represents a user, it would be much more beneficial if he could keep the user constantly involved. Buyer staffs are frequently unqualified to monitor the development of a system."

"The procedure followed is very much a function of the seriousness of the discrepancy, the extent of the effects, the point at which it is discovered, and the people involved. The customer is involved through regular written reports and status meetings."

"An effective action item system handles all discovered discrepancies. Customer lives in the building."

"Appropriate documentation is generated prior to review and review comments are incorporated subsequently consistent with configuration control concepts. The customer is involved in the evaluation of Design Review Packages and coordination of comments either at the Design Review or Technical Interchange Meeting and is involved further in design review, technical interchanges and management reviews prior to the formal review. (This project is primarily an Air Force organization project with on-site contractor assistance.)"

The respondents were asked to check a list of items that might be included in the review process. The items and the number indicating inclusion were:

Progress Reports	86%
Delivery & Computer Schedules	71
Discrepancy Report	67
Project Summaries	62
Program Listings/Documentation	62
Manpower Costs	57
Configuration Management Report	48
Computer Utilization Report	43

The respondents were asked whether the review processes were effective in their estimation for detecting and correcting significent design errors and how they might be improved. Only 40% of the respondents thought the reviews were adequately effective. Suggestions of the reasons for ineffectiveness and possibilities for improvement, with the first three items cited several times, included:

- Insufficiently detailed reviews
- Inadequate budget and schedule for review
- Greater emphasis on review during the design process
- Continuing contact between contractor and customer
- Joint generation by contractor and customer of specifications during a concept/design phase
- Pre-design meetings to establish overall conventions, concepts and interfaces prior to Part I design
- Control by a control project office over key technical aspects manned by competent, responsible personnel with authority to force correct design approach
- Adequate time for design and review
- Improved cost estimating techniques

The emphasis for improved effectiveness is seemingly concentrated on conception prior to design, close liason with the customer, and some improvement in configuration management procedures.

4.5 CONFIGURATION CONTROL PRACTICES

Configuration control is the systematic evaluation, coordination, approval and implementation of a computer program and associated documentation after formal identification of its requirements. The military have established various standard practices for configuration control and for software specification. In response to questions on this matter, just half the respondents indicated they had followed such practices and believed them an effective tool for managing the reliability and productivity of the software being developed.

Eleven projects were governed by MIL-STD's 483 and 490, six by MIL-STD-480, three by DOD Inst. 4120.17-M, two each for MIL-STD-499, NAVSHIPS 0967-011-0011, and internal programming standards, and one each for MIL-STD 1521, MIL-R 83313, and AFR 800-14. Although the military has devoted a great deal of effort to configuration management, it is not as widely practices as one might hope. Only about 25% of the respondents thought the existing standards were adequate for effective control. However, the only comment received regarding the standard practices was that they are "Too vague and subject to interpretation for effective control, especially as to the level of detail required in the specification."

5. CURRENT PRACTICES IN DATA COLLECTION

The adequacy of the software data collected is influenced by many factors ranging from the standards employed in determining the data to collect to project reluctance to release sensitive data. The respondents were first asked what standards they used and what were their major deficiencies.

5.1 STANDARD PRACTICES

First, very few projects (between 10 and 30% for various standards) used the military standard practices. However, many did have standard data that were collected for almost every aspect of software development. The list of data types and the number of projects reporting standards for collection are shown in Table 1. It would appear that the major emphasis is upon progress and schedule reporting with less than 50% of the projects using any standards for reporting other data items.

Table 1. Data Item Standards

Data Type	Percent of the Projects with Standards
Schedule Variance	76
Cost Variance	28
Project Progress	90
Problem Reports for:	
1. Design	38
2. Analysis	38
3. Integration	43
4. Code and Check	52
5. Installation	43
6. Test	43
7. Operations	24
Error Statistics	14
Manpower Utilization	52
Computer Utilization	49
Design Change Statistics	38
Software Module Statistics	33

The respondents were then asked what the major deficiencies were in the standards they used. The responses, in order of their endorsement, were:

- Inadequate for comparative study across projects 43%
- Coarseness of measures 38
- Inadequate information on project problems 38
- Failure to apply to all phase of development 29
- Inadequate information on product errors 19
- Excessive collection costs 14
- Inadequate configuration information 14
- Inadequate information on design changes 14
- Subject to interpretation; vague 14

If the allegation that data are inadequate for comparative study is true, developmental costing modules developed from them would have limited utility. Further, close management would be inhibited by coarse data, inadequate information and missing information for some phases. The other deficiencies are not greatly endorsed but indicate some dissatisfaction with current data collection practices in these areas.

5.2 MANAGEMENT CONTROL

The project managers were asked which of the data classes they used for managing their projects and where in the developmental process were the major inadequacies in the data items. Again, usage of these data was not overwhelming and very few checked all data classes. Responses were:

- Progress reports 57%
- Error reports 33
- Cost reports (including manpower and computer utilization) 33
- Change reports 24
- Problem reports 24
- Schedule 5
- Action items 5

Agreement was greater on where in the developmental process the reports were inadequate. Responses were:

- Design process 67%
- System Analysis phase 52
- Program Production 33
- Phase over 10
- Integration and Test 00

It would appear that the managers were most lacking in information during the early formative and conceptual aspects of the developmental process. Once the product was well defined, the data collected was deemed reasonably adequate. This does suggest that some attention should be directed at either greater structuring of analysis and design or that a different class of data ought to be collected. Perhaps the responses are merely an endorsement of the uncertainty that everyone feels while the system is being conceived, but a lack of good management control over the early phases of development does seem a most likely reality.

5.3 COMPARATIVE METHODOLOGY

One of the principle reasons for collecting software development data is to derive measures for evaluating and comparing different programming techniques, methods and tools. The respondents were asked to evaluate currently collected data and indicate the chief inadequacies in the data for this purpose. Their responses were:

- Project comparability 62%
- Subjectivity and bias 33
- Continuity through life cycle 29
- Reliability data 19
- Cost data 14
- Quality assurance information 10
- Standard baselines 05

5.4 COST FACTORS

The project managers were asked if they thought that state-of-the-art data collection was too costly, where the excessive cost factors lay and what could be done to reduce these costs.

In answer to where the principle costs of collecting software development data lay, the project managers said:

Volume of data	43%
Difficulty of measuring	43
Number of measures	32
Report preparation	32
Interference with work	33
Frequency of reporting	24
Reduction of data	19
Lack of defined goal	05

Additionally, there were several comments to this question:

> "The time required of production personnel in preparing reports is too great, but support personnel to do the work are too costly."
>
> "Due to the lack of clear cut goals toward which data collection can be aimed, too much unapplicable data is collected."
>
> "The preparation and maintenance of the [management] data base and the generation of documentation are both too costly."
>
> "The handling of problem and change requests is inefficient."
>
> "Manpower and the expense of indoctrination [training in data collection procedures] are excessive."

"Verification of the data is a major cost factor."

"Attempting to collect too fine grain data."

"Attempting to let the collection of data drive the system."

Costs, especially unbudgeted costs, are a source of great concern to managers although only a small proportion of the respondents said costs were a major problem in the collection of data (fourteen percent, see Section 5.1). The suggestions for improvement made by the managers provide a little more insight. These include:

- Automation
- Simplification
- Standardization
- Reduction of detail
- Better evaluation of data collection requirements

Each of these suggestions received several endorsements. There were three suggestions for automation, including the monitoring of a programming support library as a data collection technique. There were an equal number of pleas for simplification ("make it easier") with an attendant effort to sell the benefits of data collection and to explain the rationale and procedures to the project members.

The requests for simplification were also partially requests for standarized, cross-project and cross-discipline systems and procedures that could be understood by all. Another benefit foreseen for standardized and automated systems is a reduction of threatening pressure on the programmers.."Not make them feel they are being constantly overseen." Several persons felt less data collection might be the solution. One person advocated a reduction in the detail of the data collected. Another said: "Be pragmatic. Measure only what can be realistically measured." Another summed it up: "The real question is: Can the collection cost result in savings which are greater? If not, the most

cost effective solution may be simply to "wing it.""

In short, project managers would like a standard, simplified data collection system, easy to use and explain, that represents a minimum of interference in ongoing work. Whether such a system can be obtained and still provide adequate information for close management control and methodology research remains to be seen.

6. DATA COLLECTION PROBLEMS

One of the major objectives of the questionnaire was to gather experimental data on the problems associated with data collection. The project managers were asked to check off what their major problems were in collecting data. The results were:

- Subjectivity of estimates 52%
- Interference with project progress 48
- Incomparability of projects 38
- Project resistance 33
- Collection costs 29
- Invalid measures 15
- Distorted or falsified data 15
- Non-standard practices 05
- Insufficient information for valid evaluation 05
- Knowing what to collect 05
- Convincing management of the cost effectiveness of data collection 05
- Time to handle minor details 05

6.1 SUBJECTIVITY

The subjectivity of measures can arise from a variety of causes ranging from the alleged insubstantiality of the software product to personal bias on the part of those reporting. When asked what the contributing factors to the subjectivity of data collection measures were, the project managers said:

- Continual change 62%
- Optimistic bias 48
- Failure to consider all elements 43
- Lack of measurable performance 33
- Lack of "Instrumentation" 29
- Insubstantial product 10
- Innovativeness of process 05

Obviously, the project managers do not agree with the old saw that the reasons for unreliable cost estimates and poor performance are a rapidly changing technology and a logical, non-physical product. Instead they blame an unstable environment and the fallibility of estimators. Although it is not likely that human nature will change, certainly an improved data collection system should provide more objectivity and greater standardization.

The managers gave very few responses when asked how the objectivity of measures might be improved. They did suggest:

- Develop standard parameters
- Use independent audit teams
- Perform data collection research comparing a number of projects and identify the factors that contribute to the unwanted variance.

These are, of course, pertinent suggestions and are part of the justification for the Repository.

6.2 INTERFERENCE WITH PROJECT PROGRESS

When asked which major factors caused data collection to interfere with project progress, the managers checked:

- Time required to prepare reports 52%
- Distracting and irksome 33
- Interference with line of thought 10

There have been serious arguments advanced for developing more objective measures that could be taken without involving analytic and programming personnel. However, in the eyes of the managers, although data collection takes time and is irksome, it does not actively interfere with the worker's thought processes.

Suggestions for avoiding interference included:

- Automation
- Report at higher levels
- Streamline and standardize
- Use independent audits
- Place a data collector on the project staff

Some managers laid it on the line:

> "Some interference is bound to result if you are to get to the root of problems."
>
> "If data collection interferes, you are doing it wrong."

In short, quite a few managers believe that we are doing it wrong and that a more standardized, automated data collection system using non-involved data collectors is the way to go. Although formal data collection and reporting costs are quite small (estimated at 3% of project costs, see Volume 002), they may require substantially more time of analysts and programmers than the statistic indicates. More efficient collection procedures seem highly desirable.

6.3 PROJECT COMPARABILITY

When asked what project differences contribute to making data collection measures lack comparability, project managers tended to check a number of reasons. The tallies were:

- Lack of standard measures 62%
- Lack of standard techniques 57
- Software application differences 57
- Lack of standard organization 52
- Differences in management practices 48
- Technological development differences 43
- Relative "tightness" of schedules 05
- Customer requirements differences 05

In short, project managers believe that projects tend to differ on a complex of factors rather than on a single aspect. This does threaten to make methodological research difficult by making it hard to obtain an adequate sample of similar projects upon which to base an assertion. If enough of the variables are "standard," it is possible to select one or more techniques or organizations as basic variables in project comparisons.

When asked how the comparability of measures taken from different projects could be improved, project managers said:

- Develop quantitative measures of project characteristics
- Standardize the requirements of the development process
- Raise and broaden the sampling level (above the minor low-level differences)
- Use independent audit teams across projects
- Circumstances differ too greatly to achieve truly comparable projects.

Since the comparability of projects rose to the top several times in the survey, it appears to be an important problem, and one that stymies good estimates of costs, quality and project requirements. However, although the suggested solutions appear difficult to implement, they seem within the goals of the repository.

6.4 PROJECT RESISTANCE

The reluctance of project personnel to release data is an oft mentioned problem for data collection. Project members resist close monitoring as a perceived threat to their independence and professional competence and managers say that they will not release data that might help or give comfort to competitors. Less than a third of the project managers in the survey checked project resistance as a data collection problem. When asked what justification was given for project resistance, the project monitors said:

- Interference with main task 43%
- Collection costs 19
- Political Consequence 19
- "None of your business" attitude 14
- Company sensitive information 00

The results are a little surprising in that the objective reasons "interference" and "costs" are cited and the reluctance to release competative information is not cited at all. However, the perceived threat and the ways to deal with it were recognized in the suggestions for improvement:

- Do not penalize project
- Keep data sources anonymous
- Give positive assurance that honesty is desired and that corrective actions will enhance rather than hinder project productivity
- Explain the purpose of the data collection

However, in keeping with the main rationale - interference with main task - six people suggested schedule and budgetary recognition of the data collection task and making adjustments when data collection did interfere with cost.

6.5 COLLECTION COSTS

On being asked where the principle costs of collecting software development data lay, the project managers said:

- Volume of data 43%
- Difficulty of measuring 43
- Number of measures necessary 38
- Preparation of reports 38
- Interference with work 33
- Frequency of reporting 24
- Data reduction volume 19
- Lack of defined goals 05

Apparently costliness is due to a broad spectrum of factors rather than any isolated one. The amount, variety and frequency of measuring combine with the difficulty and interference effects to inflate costs that are too often bourne by technical rather than support funds and personnel.

The suggestions advanced for reducing costs include four recommendations for automatic collection, three for standardization, and two for streamlining and generalization. One person recommended better planning and one said "Don't collect any". One dissident voice suggested:

> "They [costs] probably cannot be, or should not be, reduced if any real progress is to be made in utilizing the experience of one project as the basis for expectations on another. In any case, the costs of data collection may be justified in the probable reduction of other costs."

When the managers were asked if they thought cost estimates were inflated as a resistance tactic, six agreed 'yes' and five said 'probably.' To some degree then, the managers agree that the costs of data collection may not be as great as alleged, which may account for its low rank as a problem.

6.6 VALIDITY

Despite the low number of persons citing invalid measures as a major problem for data collection, quite a few persons responded to the question asking for the major sources of error. These responses were:

- Subjectivity in measurement 48%
- Poorly defined parameters 43
- Variability from time to time, project to project 38
- Different measures project to project 33
- Insensitive measures to project difficulties 29
- Too infrequent for adequate control 14

Although subjectivity ranks high, the heavy endorsement given to the next three ranking responses indicates that standardization and consistency of the measures collected are seen as problems by many of the managers.

Suggestions to improve the validity of measures include:

- Derive and enforce standardization
- Increase programmer confidence in the purposes of data collection
- Better use of automated controls to assess the number of compilations, test runs, failures/successes, etc; in general, better instrumentation
- Comprehensive study of each project with respect to the data to be collected from personnel on the project and define standarized measures for the project
- Provide (use and pay) an independent audit team

- Concurrence on consistent definitions for measures deemed pertinent, plus objective tracking and reporting of those measures on a regular basis in terms that do not presume an intimate understanding of the package measured

6.7 FALSIFICATION

Good response was also received to the item on distorted or falsified data despite its low rank in perceived importance. In response to the question on what sort of distortions occurred, the project managers said:

• Unconscious optimistic or pessimistic bias	48%
• Deemphasis on failures	48
• Overemphasis on successes	33
• Cosmetic revision in summarization	29
• Less detail in summarization	24
• Coverups of difficulties	09
• Deliberate falsifications and half truths	05
• Poor classification	05
• Unclear definition	05
• Not considering people's reactions	05

Most interesting here is the rejection of deliberate falsification of data and the emphasis on unconscious bias. Distortions caused by filtering and summarization of data stand at midlevel.

In response to the questions on how managers might penetrate or detect falsified or distorted data, the project managers suggested:

"Through technical awareness and thorough counter and cross checks wherever possible."

"The only practical way is to be part of the project and know it as well as the technical people - which may not be really practical!"

"Use automatic tools."

"Track performance on a periodic basis and review progress with project members."

"Provide IG function with highly knowledgeable software experts from PMO staff (not corporate headquarters.)"

"Allow the data supplier to be completely free to provide the data without reprimand or reward. Cross-check data by data collection in some other fashion."

"Success depends upon having a very clear understanding of the need, the requirements, and user biases. Unless a project is aiming at clearly delineated performance characteristics almost any assessment is equally valid."

"Know what you are doing. If you do, falsehoods quickly stand out."

"Avoid distortions by good planning."

Apparently the project managers place their faith in technical competence reinforced by independent audits and cross-checks. Recognition is given to psychological factors, but without strong emphasis.

7. HAWTHORNE EFFECTS

It has been alleged that the very act of measuring performance can influence that performance favorably. The project managers were asked whether they thought that data collection could enhance the reliability and excellence of project work. There was moderate agreement with this statement according to the responses:

- Encourages taking greater care 43%
- Makes workers aware of inadequacies 38
- Makes job seem important; gives recognition to workman 33
- Makes worker act more deliberately 05

8. TECHNOLOGICAL ADVANCE

The project managers were asked what developments they saw in the data-processing state-of-the-art that might improve or hinder the process of collecting software development data. They said:

	HELP	HINDER
• Automated data collection system	67%	05%
• Automated management tools	57	00
• Greater standardization of language and tools	52	05
• Improved measurement techniques	43	05
• Proofs of correctness methodology	24	19
• Structured programming	24	05
• Software engineering techniques	24	10
• More analytic devices for software development	24	00
• Increased emphasis on reliability	14	14

In short, greatest hope is seen to lie in greater automation and better, standardized measures. Although not felt generally, both proofs of correctness and the greater emphasis on reliability are seen as adding somewhat to the problems of data collection.

9. SUMMARY

In a survey consisting largely of programming project managers and system program offices, a certain polarization of opinion might be expected. However, there is a wide spread of opinion on the sensitivity of all kinds of data items. Most sensitive are cost data and, more so, data that directly reflects project performance evaluations such as cost and schedule variances. Least sensitive are objective statistics about resources, program modules, modifications, problems and errors that do not reflect cost or performance evaluations.

In reference to the software development process, the managers felt that planning, analysis, and review in the early developmental phases were inadequate and that insufficient information exists or is collected about these phases. In later phases, independent test teams are not used as often as one might desire (only 30% of the time) and data is not often collected during the operational phase of a system. Although many projects (approximately 50%) were governed by the MIL-STD's only about 25% thought these standards were adequate. Most projects (90%) used some sort of progress report and 75% used schedule variance for project control. No other data were used by more than 50% of the projects. Hence, one might conclude that very few projects "control by the numbers" from which it might be concluded that most project managers control their projects by personal supervision. On the other hand, the dissatisfaction of the managers with the variations in data standards and procedures from project to project may make them distrustful of what the collection data may tell them.

MISSION
of
Rome Air Development Center

RADC plans and conducts research, exploratory and advanced development programs in command, control, and communications (C^3) activities, and in the C^3 areas of information sciences and intelligence. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.