IDENTIFICATION AND EVALUATION OF SOFTWARE MEASURES

David N. Card

COMPUTER SCIENCES CORPORATION

and

GODDARD SPACE FLIGHT CENTER

SOFTWARE ENGINEERING LABORATORY

Prepared for the

NASA/GSFC

Sixth Annual Software Engineering Workshop

INTRODUCTION

The purpose of this presentation is to describe and demonstrate a large-scale, systematic procedure for identifying and evaluating measures that meaningfully characterize one or more elements of software development. The background of this research, the nature of the data involved, and the steps of the analytic procedure are discussed. The presentation concludes with an example of the application of this procedure to data from real software development projects.

As the term is used here, a measure is a count or numerical rating of the occurrence of some property. Examples of measures include lines of code, number of computer runs, person-hours expended, and degree of use of top-down design methodology. Measures appeal to the researcher and the manager as a potential means of defining, explaining, and predicting software development qualities, especially productivity and reliability.

Measures may be classified into four groups as illustrated by the software development model presented in Figure 1. It shows these components: a problem, a solution-generating process, the environment in which that process takes place, and the solution (or software product). Measures can be employed to characterize the components of this model and to show their interrelationships. Some examples of appropriate measures for each component are also shown in the figure.

The Goddard Space Flight Center (GSFC) Software Engineering Laboratory (SEL) is engaged in an effort, part of which this presentation describes, to develop a concise set of such characteristic measures. The SEL and its activities are discussed in more detail in Reference 1.

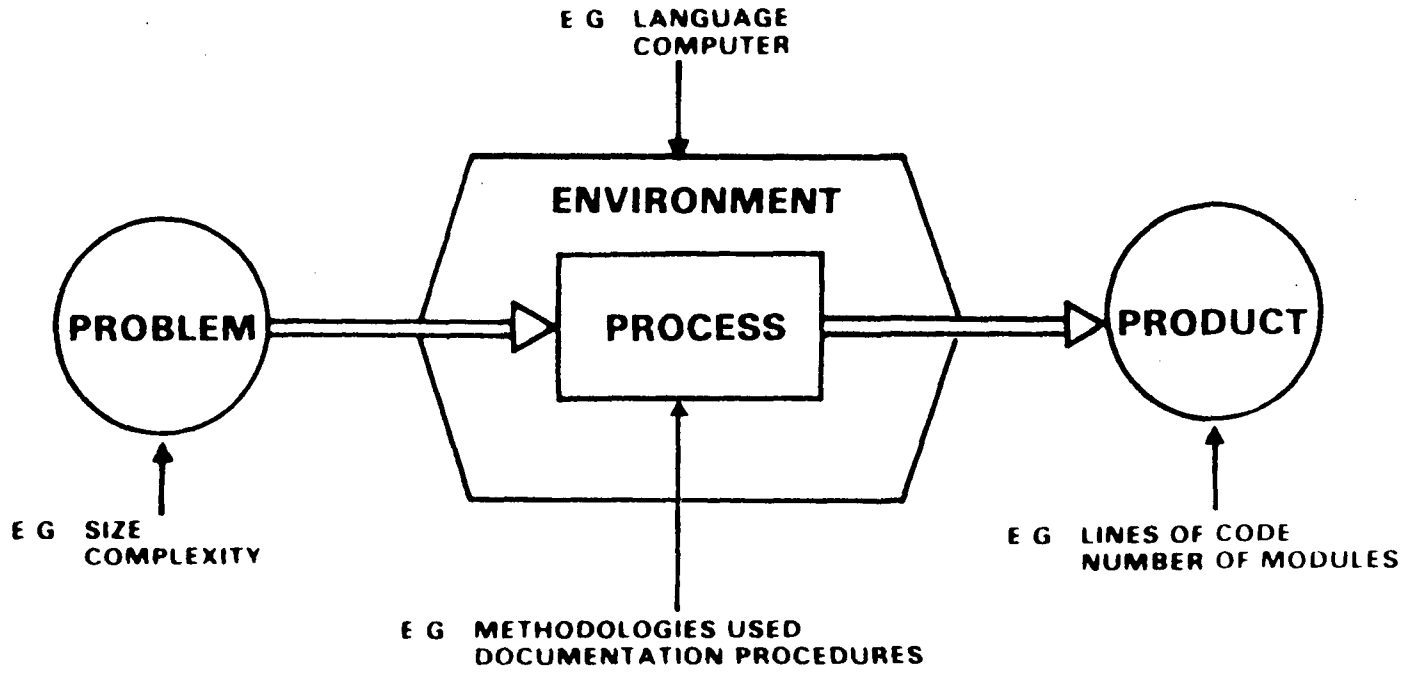


Figure 1. SEL Software Development Model

The approach to software measurement adopted in this presentation is different from that generally followed. The usual procedure is to select high-level "qualities" and then to seek numerical criteria or measures of these qualities. McCall (Reference 2) has developed a comprehensive system of such qualities and appropriate measures. However, the goal of the approach followed here is to identify the qualities being measured by the data collected rather than to attempt to associate measures with previously specified qualities. The measures considered in this analysis are described in the next section.

DATA DESCRIPTION

Clearly, the number of potentially useful measures is large; the SEL has selected more than 200 for study. These measures cover the entire range of software development activity as experienced by the SEL. However, the analysis described here will focus on the relationships among measures of the process and product components of the software development model (see Figure 1).

Therefore, a data subset containing only the 60 measures relevant to those two components was used. The measures (or variables) used are listed in Table 1 (see Appendix A). This list does not necessarily exhaust the possibilities for measures in those areas; however, this group of measures is believed to form a comprehensive set. The process measures class is represented by three subclasses: methodology (Table 1a), tools (Table 1b), and documentation (Table 1c). Note that the methodology class is further subdivided by development phase into design, code, and test measures. The product class (Table 1d) includes size and resource measures.

The data used in this analysis were collected by the SEL from 22 actual medium-scale, scientific software development projects. Values for all these measures were determined for each project. The values are ratings of the degree of use, counts, or rates per line of code, as indicated in Table 1. Degree-of-use process measures are expressed as relative scores on a scale from zero to five. The exact derivation of these scores will be explained in a forthcoming SEL document (Reference 3).

ANALYTIC PROCEDURE

The 60 measures just described are not unique or independent. Some may, in fact, measure the same or related qualities. The object of the analytic procedure is to identify the most basic set of qualities (or properties) being measured by the group of 60. A "basic" quality is defined to be one that is independent of all other such qualities. This subset, then, defines the basic quality characteristics describing the projects from which the data were obtained.

The procedure to be proposed is "large scale." That is, it is appropriate when a large number of measures (or variables) are to be evaluated. The researcher interested in studying the relationships of only a few specific measures can probably get better results from regression and hypothesis testing techniques. Nevertheless, this procedure can be useful as a screening tool for detecting confounding effects in the data before selecting other statistical techniques.

The analytic procedure followed in this experiment has two steps, as indicated in Figure 2. These are the application of a test of normality to the candidate measures (data), followed by a factor analysis of those not rejected by the test. The result of this procedure is a descriptive, rather than a predictive, model of the data. The procedure identifies the descriptive factors common to the set of measures. Thus, the original measures are organized into a number of groups (or factors) smaller than the number of measures input to the procedure. These factors correspond to the basic qualities sought for in the data. The steps of this procedure are discussed in more detail in the following sections.

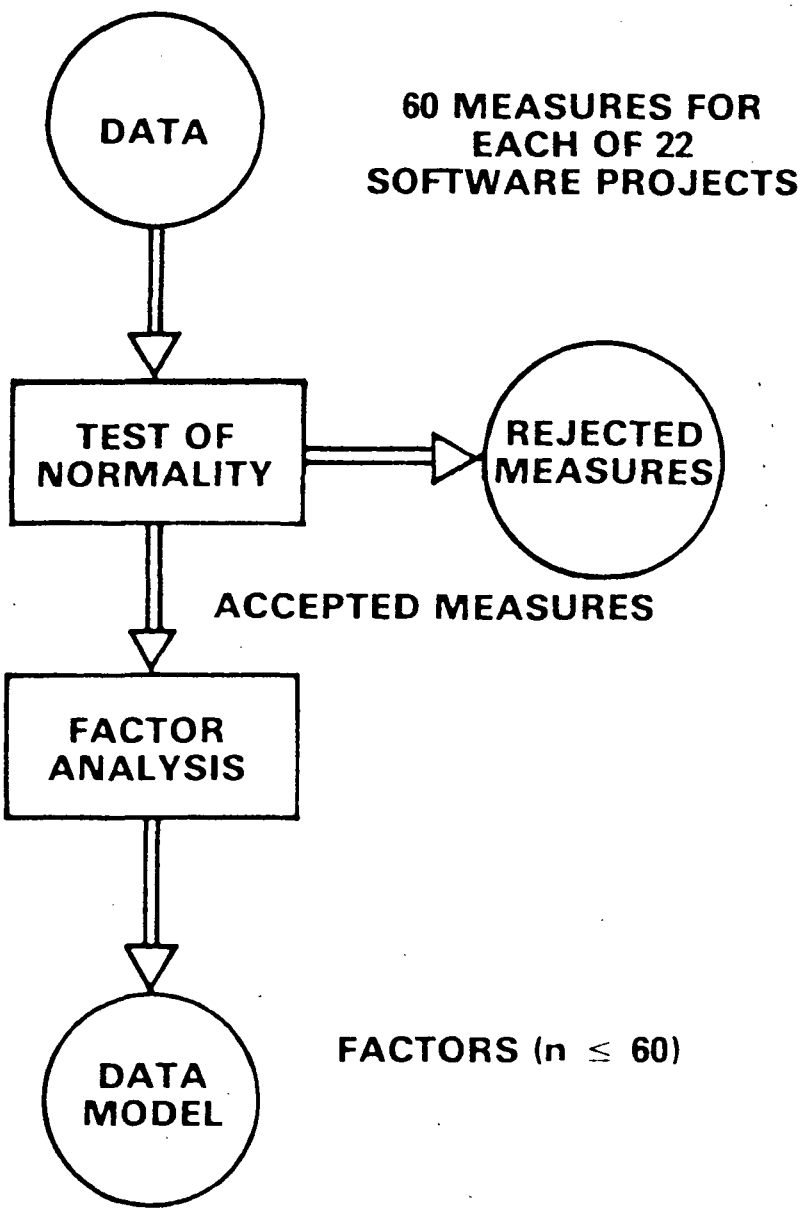


Figure 2. Analytic Procedure

TEST OF NORMALITY

The test of normality analyzes the probability distribution of a measure. The observed values of each measure are distributed over some range. The normal distribution is readily identifiable in Figure 3. The test of normality will detect measures whose values are distributed in a pattern significantly different from the normal. For example, it would reject a measure with values clustered at one end of the range (skewed) rather than distributed symmetrically across it.

This is not a very powerful test. It will accept any approximately symmetrical distribution even if that distribution is not truly normal. However, the test is important because approximate normality of the data is an assumption of step two, the factor analysis.

Six measures from the set of 60 were rejected by the test of normality using the 0.05 level of significance. These are measures of techniques for which insufficient examples of use were available. Consequently, most projects had scores of zero for these degree-of-use measures, a result that produced dramatically skewed distributions. They are

- HIPO Design Technique
- Verification and Validation Team (two measures)
- Requirements Language Tool
- Configuration Management Tool
- Unit Development Folders

These measures could, however, be used in some other types of analyses not considered here.

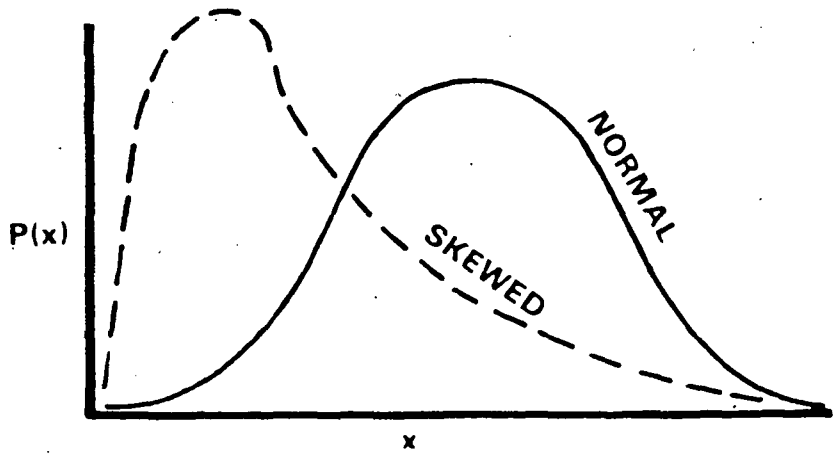


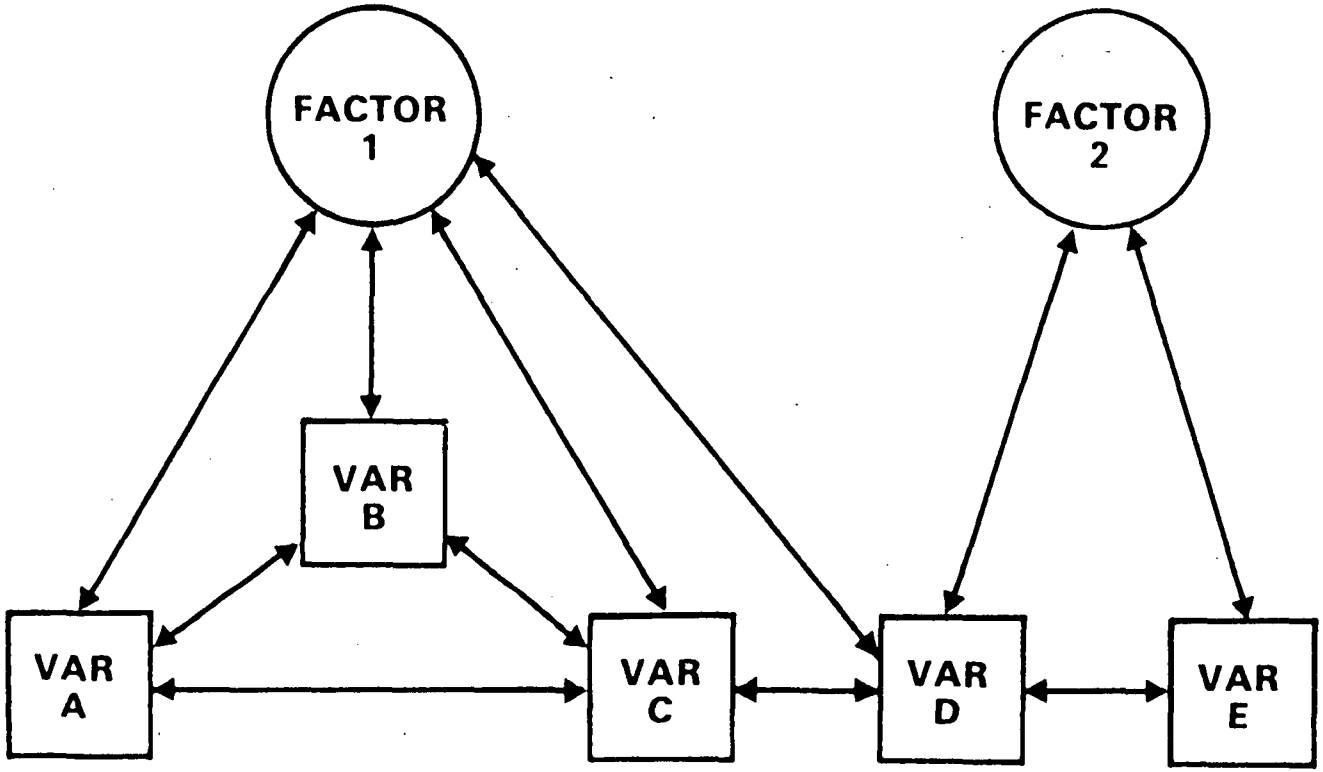
Figure 3. Test of Normality

FACTOR ANALYSIS

The 54 remaining measures were included in the factor analysis. The goal of the factor analysis is to "discover" the underlying structure of the data. Factor analysis hypothesizes the existence of a set of statistically independent "factors" that are not directly measurable by the experimenter. Measures (or variables) are the quantities that are observed in practice. However, the apparent correlations among measures can be interpreted to be due to their joint correlation with common factors (see Figure 4). That is, two or more measures correlated with the same factor will be correlated with each other. The desirable result of a factor analysis is the extraction of a smaller set of factors whose relationships are known (they are independent) from the larger set of measures whose relationships are more complex.

Consider this example of the factor concept. The number of errors in a piece of software and its mean time to failure are measures related to reliability and are correlated with each other. However, neither measure by itself is a full description of reliability. Such things as the location of the error and the severity of the failure must also be considered. Therefore, the reliability quality factor is not directly measurable although a number of measurable variables are correlated with it.

A successful factor analysis will explain such groups of related measures. Thus, each factor defined will correspond to a distinct basic quality being measured by the original set of variables. These qualities are the sources of variation (or differentiation) among the projects studied.



NOTE: VARIABLES MAY BE CORRELATED.
FACTORS ARE INDEPENDENT.

Figure 4. Concept of Factor Analysis

The principles of factor analysis are explained in detail in the text by Harman (Reference 4). A number of software implementations of factor analysis are available. The specific software used in this analysis was the principal components factor procedure of the Statistical Analysis System (Reference 5).

SUMMARY OF RESULTS

Further analysis of the 54 process and product measures that passed the test of normality produced a factor model containing 5 factors that explained 77 percent of the variance of the original measures. The meaning of each factor is determined by examining the measures that are closely correlated with it. These factors and the amount of variance accounted for by each are as follows:

- Methodology Intensity (31%)
- Project Size (25%)
- Computer Usage (9%)
- Quality Assurance (8%)
- Change Rate (5%)

The variance associated with a factor is a measure of the degree to which that factor differentiates among the projects (or cases) studied. Thus, it is a measure of information content. A larger portion of the total variance could have been accounted for by using a larger number of factors. The relationship of the number of factors to the variance explained by the factor model is illustrated in Table 2 of Appendix A. The interpretation of additional factors is difficult because none of the original measures are highly correlated with them. Therefore, they are not included in this preliminary definition of the factor model.

The correlations of the original measures with the five factors are listed in Table 3 of Appendix A. Only correlations greater than 0.526 (the 0.01 level of significance) are reproduced. The measure showing the highest correlation with a factor can be taken as the best estimator of that quality factor from among the original measures included in the analysis. These "best" estimators are indicated by asterisks in the tables.

Remember that, although the factors are mutually independent, any given measure may be correlated with more than one factor and/or with other measures. The factor model does, however, identify the strongest relationships in the data. Some specific observations are made below about each of the factors defined by the analysis.

Factor 1 - The first and most powerful factor (Table 3a in Appendix A) is highly correlated with degree-of-use process measures; thus, this factor may be interpreted to represent the degree to which formal methodology was applied during development. The most strongly correlated measure, methodology reinforcement (the extent to which adherence to specified methodologies was enforced by management), supports this interpretation. The strong correlation of so many methodology, tool, and documentation measures with a common factor suggests that simple regression and hypothesis testing techniques are inappropriate for analyzing such effects because of their inability to isolate the action of a single technique from among the actions of other techniques.

Factor 2 - The second factor (Table 3b in Appendix A) is clearly related to the size of the development effort and product. Its "best" estimator is person-hours. The correlation of top-down coding with this factor illustrates the descriptive, rather than predictive, nature of factor analysis. The proper conclusion based on this observation is that more top-down coding tends to be used in small projects than in large ones, not that top-down coding necessarily reduces the size of a development effort.

Factor 3 - The third factor (Table 3c in Appendix A) contains a number of measures related to the pattern of computer usage. This factor indicates that the manner and degree of computer usage reflect the use of certain development tools and techniques. The "best" estimator of this factor is top-down design.

Factor 4 - The fourth factor (Table 3d in Appendix A) has only one measure, semiformal quality assurance, significantly correlated with it. Thus, its meaning is difficult to establish. However, a substantial amount of variance (8 percent) is associated with this factor. The preceding factor contained five variables but explained only slightly more variance (9 percent). Thus, this factor and measure deserve closer examination in future analyses.

Factor 5 - The last factor (Table 3e in Appendix A) clearly describes the change rate. The interpretation of this factor is important since, as a consequence of the mutual independence of factors, it is independent of the four factors previously defined. Hence, methodology intensity, project size, and computer usage do not appear to be related to each other or to code stability (reliability), as measured by the change rate.

Another feature of this model should be noted. Although productivity was most strongly correlated with factor 4, it was not significantly correlated with any factor. Productivity may still be related to specific methodologies but not to the general factors just defined. Thus, the information provided by this procedure about productivity and reliability is negative in this example because unrelated qualities and measures were identified rather than related ones.

CONCLUSION

The results presented here are preliminary. Conclusions based on the factor model just developed may change as more data become available and as the procedure is refined. However, the analysis has demonstrated its capacity to resolve some important questions about the data. The conclusions are as follows: the basic qualities being quantified by the original measures can be identified and enumerated; their relative importance or strength (in terms of percentage of variance accounted for) can be established; and a "best" estimator can be selected for each quality.

Therefore, we can define a concise set of quality measures that meaningfully characterizes the process and product components of the software development model and that can serve as a framework for further research. These qualities and associated measures can be studied in greater detail with other techniques to determine their relationships to productivity and reliability more exactly. Hence, these results are a first step toward defining, explaining, and predicting software reliability and productivity in the SEL environment.

APPENDIX A - SUMMARY OF FACTOR ANALYSIS

This appendix consists of a series of three tables that summarize the factor analysis procedure described in the preceding discussion. Table 1 describes the measures evaluated in this analysis. Table 2 identifies the variances associated with factors. Table 3 lists the significant correlations (at the 0.01 level of significance) of measures with factors.

Table 1a. Methodology Measures

(DEGREE OF USE)

ORGANIZATION	—	CHIEF PROGRAMMER
DESIGN	—	WALKTHROUGHS
DESIGN	—	FORMAL REVIEWS
DESIGN	—	FORMALISMS
DESIGN	—	TREE CHARTS
DESIGN	—	PROGRAM DESIGN LANGUAGE (PDL)
DESIGN	—	HIERARCHICAL INPUT PROCESSING OUTPUT (HIPO)
DESIGN	—	TOP-DOWN
DESIGN	—	ITERATIVE ENHANCEMENT
CODE	—	STUBS
CODE	—	TOP-DOWN
CODE	—	STRUCTURED
CODE	—	WALKTHROUGHS
CODE	—	READ
CODE	—	CONFIGURATION CONTROL
TEST	—	FORMALISM
TEST	—	FOLLOWTHROUGH
TEST	—	BATCH
TEST	—	V&V PRESENCE
TEST	—	V&V USE

Table lb. Tools Measures

(DEGREE OF USE)

FORMAL TRAINING IN METHODOLOGY
INFORMAL TRAINING
METHODOLOGY REINFORCEMENT
REQUIREMENTS LANGUAGE (MEDL-R)
DESIGN LANGUAGE (PDL)
PRECOMPILER (SFORT)
SOFTWARE AIDS (e.g., EXREF, MAP, LIST)
LIBRARIAN
DATA GENERATORS
TERMINALS (TSO)
REMOTE JOB PROCESSING (RJP)
CONFIGURATION ANALYSIS (CAT)

Table 1c. Documentation Measures

(DEGREE OF USE)

SEL FORMS
DESIGN DOCUMENT
DESIGN DECISIONS
SEMIFORMAL QUALITY ASSURANCE
ACTIVITY NOTEBOOKS
UNIT DEVELOPMENT FOLDERS
TEST PLANS
USER'S GUIDE/SYSTEM DESCRIPTION
FORMAL TREATMENT OF USER'S GUIDE
WEEKLY/MONTHLY PROGRESS REPORTS

Table 1d. Resource/Product Measures

(COUNTS AND RATES)

NUMBER OF COMPONENTS
TOTAL MODULES
NEW MODULES
MODIFIED MODULES
TOTAL LINES OF CODE (INCLUDES COMMENTS)
NEW LINES OF CODE (INCLUDES COMMENTS)
MODIFIED LINES OF CODE
NUMBER OF COMPUTER RUNS
NUMBER OF CHANGES
PAGES OF DOCUMENTATION
TOTAL MANHOURS
TOTAL COMPUTER HOURS
PERCENT OF NEW CODE
CHANGES PER LINE OF CODE
CHANGES PER LINE OF NEW CODE
NEW LINES + 20% OF REVISED LINES
LINES OF CODE PER MANHOUR
COMPUTER HOURS PER LINE OF CODE

Table 2. Preliminary Eigenvalues and Variances Associated With Factors

FACTOR	1	2	3	4	5	6	7	8	9	10	11
EIGENVALUES	16.492786	13.305087	4.744286	4.063959	2.855640	2.438981	1.738979	1.555128	1.469211	1.101198	0.931850
PORTION	0.305	0.246	0.088	0.075	0.053	0.045	0.032	0.029	0.027	0.020	0.017
CUM PORTION	0.305	0.552	0.640	0.715	0.768	0.813	0.845	0.874	0.901	0.922	0.939

FACTOR	12	13	14	15	16	17	18	19	20	21
EIGENVALUES	0.685056	0.621503	0.495917	0.427863	0.397183	0.255911	0.209853	0.153974	0.055635	0.000000
PORTION	0.013	0.012	0.009	0.008	0.007	0.005	0.004	0.003	0.001	0.000
CUM PORTION	0.952	0.963	0.972	0.980	0.987	0.992	0.996	0.999	1.000	1.000

NOTE: Only five factors were retained in the analysis.

Table 3a. Factor 1

MEASURE	CORRELATION
CHIEF PROGRAMMER ORGANIZATION	.62
DESIGN WALKTHROUGHS	.75
FORMAL DESIGN REVIEWS	.75
DESIGN FORMALISMS	.83
DESIGN TREE CHARTS	.65
PROGRAM DESIGN LANGUAGE (METHODOLOGY)	.63
CODE STUBS	.86
CODE WALKTHROUGHS	.69
CODE READING	.60
CONFIGURATION CONTROL (METHODOLOGY)	.62
TEST FORMALISMS	.74
TEST FOLLOWTHROUGH	.72
FORMAL TRAINING IN METHODOLOGY	.78
INFORMAL TRAINING	.61
METHODOLOGY REINFORCEMENT	.89*
DESIGN LANGUAGE (TOOL)	.64
SOFTWARE (CODING) AIDS	.68
LIBRARIAN	.85
DATA GENERATORS	.71
REMOTE JOB ENTRY	.54
SEL FORMS	.76
DESIGN DOCUMENT	.73
DESIGN DECISION (DOCUMENTATION)	.75
ACTIVITY NOTEBOOKS	.76
USER'S GUIDE/SYSTEM DESCRIPTION	.69
WEEKLY/MONTHLY PROGRESS REPORTS	.69

NOTE: VARIANCE ACCOUNTED FOR: 31%.

Table 3b. Factor 2

MEASURE	CORRELATION
NUMBER OF COMPONENTS	.89
TOTALS MODULES	.89
NEW MODULES	.85
MODIFIED MODULES	.80
TOTAL LINES	.91
NEW LINES	.92
MODIFIED LINES	.77
NUMBER OF RUNS	.91
NUMBER OF CHANGES	.93
PAGES OF DOCUMENTATION	.94
PERSON HOURS	.96*
COMPUTER HOURS	.88
DELIVERED LINES	.93
TOP-DOWN CODING	−.56

NOTE: VARIANCE ACCOUNTED FOR: 25%.

Table 3c. Factor 3

MEASURE	CORRELATION
COMPUTER HOURS/LINES OF CODE	.60
TOP-DOWN DESIGN	.88*
BATCH TESTING	.70
REMOTE JOB ENTRY	.69
TEST PLANS	−.57

NOTE: VARIANCE ACCOUNTED FOR: 9%.

Table 3d. Factor 4

MEASURE	CORRELATION
SEMIFORMAL QUALITY ASSURANCE	.60*
(PRODUCTIVITY	−.30)

NOTE: VARIANCE ACCOUNTED FOR: 8%.

Table 3e. Factor 5

MEASURE	CORRELATION
CHANGES/LINES OF CODE	.73*
CHANGES/LINES OF NEW CODE	.64

NOTE: VARIANCE ACCOUNTED FOR: 5%.

REFERENCES

1. Computer Sciences Corporation, CSC/TM-81/6104, The Software Engineering Laboratory, D. N. Card, et al., October 1981

2. Rome Air Development Center, RADC-TR-77-369, Factors in Software Quality, J. A. McCall, P. K. Richards, and G. F. Walters, November 1977

3. Computer Sciences Corporation, Evaluation and Application of Subjective Measures of Software Development, D. Card and G. Page (in preparation)

4. H. H. Harman, Modern Factor Analysis, Chicago: University of Chicago Press, 1976

5. J. T. Sall, et al., Statistical Analysis System User's Guide, SAS Institute, 1979