Edgar H. Sibley
Panel Editor

Computer-aided software engineering (CASE) is moving into the problem-solving domain of the systems analyst. The authors undertook a study to investigate the various functional and behavioral aspects of CASE and determine the impact it has over manual methods of software engineering productivity.

CASE Productivity Perceptions of Software Engineering Professionals

Ronald J. Norman and Jay F. Nunamaker, Jr.

As we move closer to the 1990s, business and scientific software engineering workbench tools are becoming a pervasive market commodity. Perhaps the most notable methodology that is being supported by these automated tools is the structured methodology and its many variants. Referred to as CASE, these automated tools represent many years of research on integrated development environments (IDE) [5, 8, 9, 12, 25]. CASE can be viewed as an environment that supports the software engineering process.

It is estimated that thousands of medium and large enterprises are using CASE products as part of their system-building process based on the intuition that this is the way to go for improved productivity and system quality. Over the last thirty years, software engineering has focused more on the software that is closest to the machine such as compilers, operating systems, and database management systems, but is now moving rapidly into the problem solving domain of the software engineer (systems analyst).

Recent CASE advertisements suggest that users are reporting productivity improvements ranging from 30 to 300 percent; however, we are not aware of any empirical studies in the research literature that investigate CASE technology or its effects on productivity. Our automated workbench tool-support research dates back to PSL/PSA [18,19] and SODA [15], and the motivation for this research comes primarily from the discussions we have had with MIS directors and managers who must make the decision to embrace current generation CASE technology or continue looking beyond these CASE products. This study was undertaken to investigate which functional and behavioral aspects of CASE technology, from the software engineer's point of view, contribute the most favorably toward increasing their productivity over comparable manual methods.

THE RESEARCH METHODOLOGY

This study focuses specifically on the perceptions of management information systems (MIS) professionals, who perform systems analysis functions using CASE technology. We will refer to this group as software engineers throughout the study. The results of these perceptions will allow us to make some inferences and observations about specific functional parts of CASE technology as well as the effectiveness of CASE technology to enhance the communicative and standardizational aspect of information systems development efforts.

Since there is a dearth of research on the productivity of software engineers, several studies that investigated programmer-productivity techniques and tools were reviewed [4, 6, 14, 20, 26]. One study [4] specifically investigated prioritization of tools using programmer perceptions and is foundational to this study.

The study we will describe was performed to determine the ordering and underlying relationships of CASE technology. A preference map of similarity ranking was constructed using a psychometric scaling method called multidimensional scaling (MDS) [10, 17, 21, 22]. Psychometric scaling methods are used for measurements of mental traits, abilities, and processes. The method is called "metric" because it requires psychological estimates of metric distances between the stimuli [17]. It can help systematize data in areas where organizing concepts and underlying dimensions are not well developed [16]. MDS is a useful mathematical tool that enables us to represent the similarities of objects spatially as in a map.

Basically, the MDS technique endeavors to place n stimuli in a k-dimensional euclidean space ($k < n$) such that all pairwise similarities between stimuli are preserved. In other words, if the similarity between two stimuli has stimuli j and l more similar than j and k, then the corresponding distance in an MDS space between stimuli j and k should be greater than that distance between stimuli j and l [4]. This configuration reflects the hidden structure in the data and often makes the data much easier to comprehend [10]. Sometimes, however, structure can be observed in the multidimensional space in addition to or instead of that provided by dimensional interpretation. Neighborhoods or regions of the space may have meaning. Guttman [3] argued that a neighborhood or pattern approach is preferable to the traditional dimensional approach. One way to locate or interpret neighborhoods involves the application of hierarchical clustering [1, 23, 24]. The clusters can be drawn in the multidimensional space as loops around the relevant stimulus points. Once this is done, we can seek some characteristic common to the objects in a cluster. Usually this is done subjectively, as an act of creative interpretation. Since MDS is almost always used as a descriptive model for representing and understanding the data, other considerations enter into decisions about the appropriate dimensionality such as interpretability, ease of use, and stability [10].

In the present study, subjects ranked pairs of CASE product functions (i.e., data flow diagrams, structure charts, presentation graphics, etc.) in terms of how they perceived the similarity of each one affecting their productivity. Two additional factors which were considered very important during system development were also included along with the CASE product functions. They were:

(1) communication among project team members, and
(2) adherence to the enterprise's system development standards.

For purposes of this study, productivity was left undefined so that the respondents' perceptions would be based on their own definition of productivity. The pairwise ranking allows the MDS technique to construct a space of a smaller number of dimensions than the total number of stimuli from which the pairwise rankings were obtained. In this way, the stimuli are mapped onto a smaller set of features that the subjects may have used to make their judgments about all of the stimuli. The underlying features of this space can be suggested from the relationships that exist in the space and the clusters that associate with a particular dimension.

The 91 subjects were all using CASE technology in performing software engineering tasks. Sixty-seven percent of the target systems being analyzed by these subjects were scheduled to use either COBOL or a 4th-generation language (4GL). The remaining target systems were to be developed using C (8 percent), Pascal or BASIC (0 responses), "other" (23 percent), or "no language" (2 percent). The subjects were from 47 different enterprises across the U.S. and Canada representing over a dozen standard industry codes (SIC), a wide variety of enterprise sizes, and MIS budgets.

TABLE I. Survey Stimulus Items (in alphabetical order)

1.	Analysis → Graph Analysis
2.	Analysis → Entity List
3.	Analysis → Report Writer
4.	[CASE product] works on both PC and mainframe
5.	Data Dictionary
6.	Data Flow Diagram (Gane & Sarson, Yourdon)
7.	Entity/relationship data model (Chen or MERISE)
8.	Import and/or Export Facility
9.	LAN support
10.	Logical Data Model diagram (IBM)
11.	Presentation Graphics
12.	Project member's communication via [CASE product]
13.	Project standardization
14.	Record Layout Generation
15.	Screen/Report Design
16.	Structure Charts (Constantine)
17.	Structure Diagrams (Jackson)

A stimulus list of 15 technological functions of one CASE product was prepared, and the two behavioral functions mentioned earlier were added to the list. This is illustrated in Table I. All subjects were users of EXCELERATOR, a leading, commercially available CASE product in order to hold this part of the research constant. Although the implementation of technological functions may vary in other CASE products, most of the competing CASE products support an equivalent function. Testing of all 17 functions required that 136 paired comparisions be made by the subjects.

For the survey, we chose a personal computer (PC)-based survey instrument. It was designed to capture perceptions of the subjects' productivity with the use of CASE technology compared to manual methods as shown in Figure I. This approach yielded a higher completed survey response rate than traditional paper-based surveys. Prior to administering the actual PC survey, a prototype was tested at an annual user's conference, and adjustments were made to the instrument. Ninety-nine subjects started the actual survey and 91 completed all 136 comparisons. Subjects could request on-line help definitions (Appendix A) of the presented pair of stimuli at any time during the survey. Complete details of the research methodology are reported in [13].

RESULTS

The method of paired comparisons is usually assumed to yield a more reliable ordering than that obtained by requiring a respondent to order a whole group of objects directly [2]. Checking for outliers (i.e., consistency of a subject's responses) can be done by calculating the number of inconsistent triads among a subject's responses and used to define a coefficient of consistence [7]. In so doing, we found that all 91 subjects were responding with a consistent pattern that was significantly better than chance. The two primary results were a dominance ranking for the CASE functions and a cluster analysis of the functions based on perceived

Of the following items which one most increases your productivity over manual methods:

a. Data Flow Diagrams
b. Structure Charts

Enter a or b?

Rate how similar these items are in their effect on your productivity (1–7, 7 = very different)?

Enter 1 to 7 ?

FIGURE 1. Sample of the Survey's Two Questions

productivity ratings.

Dominance Rankings

The rank order of the 17 stimuli (based on 12,376 choices) according to the preference of the 91 subjects is shown in Table II. The top-ranked stimulus, Data Flow Diagram, comes as no surprise, as the manual use of this tool to draw and revise requirements has been viewed as extremely time consuming [27]. The establishment and maintenance of a Data Dictionary to support either data flow or data modeling techniques requires significant manual labor and may account for its placing second in the results. Adherence to enterprise system development standards, Project Standardization in Table II, ranked a surprising third. We consider this to be a very significant result since most CASE products are sold (and purchased) primarily on their ability to allow software engineers to produce models of the user's requirements. Therefore, CASE technology may enforce enterprise systems development standards in an unobtrusive way. The fourth ranked stimulus, screen/report design, may indicate a high use of prototyping by the subjects, and CASE technology affords them a significant productivity improvement in developing screens and reports.

This rank ordering, however, only reveals part of the picture, since these stimuli can be chosen over one another based on more than one attribute. For example, although Data Flow Diagram and Data Dictionary are generally preferred over all other CASE functions, the Data Flow Diagraming function, on a dimension of completeness and consistency checking, may be considered less important than, for example, the three Analysis functions that perform these functions. The ranking order could change depending on how general or how refined the question. The next analysis attempts to unfold these preferences in a more complete way by using the implied features and attributes the subjects may have been using to make their selections.

Multidimensional Scaling and Cluster Analysis of CASE Functions

As described earlier, the rankings of similarity can be decomposed into a number of dimensions that represent all the original rankings. This can be done by plotting each of the 136 choices on a visually interpretable graph. Since inconsistent triads may occur, plotting each of the 136 choices on a single dimension would result in an intransitivity. This means that the three stimuli could not be arranged linearly and preserve the pairwise preferences. Adding a second dimension will eliminate this intransitivity, as one of the stimuli can be placed in the two-dimensional plane at an appropriate distance between the other two. In this way, MDS attempts to accommodate all possible 136 ranks and potential intransitivities in first one dimension, then two, three, or more.

For the software engineer's preferences, the preferred representation of the 136 similarity rankings was in three dimensions with 88 percent of the total original pairwise ranks accounted for. The SPSS-X ALSCAL MDS program[1] was used, and Table III presents the preferred three-dimensional solution.

The representation of these similarity judgments is presented in Figure 2. Each axis has a suggested interpretation identifier which is based on the coordinate position of each of the stimuli and the clusters of stimuli that were determined from a hierarchical cluster analysis [1, 23, 24]. SPSS-X's CLUSTER program, which uses Ward's [24] clustering method, was used. Although the clustering was performed using the stimuli points on all three dimensions, for visual clarity, these clusters are only shown in the plane of Figure 2 as an area circumscribing the appropriate subset of functions. Those functions that lie below the plane are represented by upside-down, dotted-line flag poles.

MDS advocates claim that it is not essential that all axes be labeled, and they caution against arbitrary axis labeling [10, p. 49]. In analyzing the resulting MDS

[1] For more information on the SPSS-x ALSCAL MDS and SPSS-X CLUSTER program please contact SPSS Inc., 444 North Michigan Avenue, Chicago, IL 60611.

TABLE II. CASE Functions Ranking (n = 91)

Stimulus item	Number of times selected	Pct. selected	Relative choice
Data Flow Diagram	1155	0.79	1.00
Data Dictionary	1128	0.77	0.98
Project Standardization	862	0.59	0.75
Screen/Report Design	857	0.59	0.74
Presentation Graphics	854	0.59	0.74
Analysis → Report Writer	827	0.57	0.72
Analysis → Entity List	728	0.50	0.63
Entity/Relationship data model	726	0.50	0.63
Structure Charts	721	0.50	0.62
Logical Data Model	712	0.49	0.62
Analysis → Graph Analysis	683	0.47	0.59
Project member's communication via [CASE product]	616	0.42	0.53
Structure Diagrams	602	0.41	0.52
Record Layout Generation	598	0.41	0.52
Import and/or Export Facility	577	0.40	0.50
[CASE product] works on both PC and mainframe	453	0.31	0.39
LAN support for the CASE product	277	0.19	0.24
Total Choices	12,376		
Total Choices Per Item	1,456		

TABLE III. SPSS-X ALSCAL MDS 3-Dimensional Solution

Stress = 0.135 RSQ = 88.3%

Configuration Derived in 3 Dimensions: Stimulus Coordinates

Stimulus number	Stimulus name	Plot symbol	Dimension 1	Dimension 2	Dimension 3
1	DFD	DFD	1.0744	−1.3251	−0.2477
2	STRUCCHT	SC	0.8922	0.3819	0.2576
3	DATA-MODL	DM	0.2790	−0.1629	1.0902
4	ERMODEL	ER	1.3073	−0.3452	1.1931
5	STRUCDIA	SD	0.1377	0.4369	0.9732
6	PRESGRAP	PG	1.4321	1.3182	−0.5932
7	ANAGRAPH	AG	0.4011	−0.4816	0.6679
8	RECLAYOT	RL	−0.0236	1.6953	0.2209
9	DATADICT	DD	1.0707	−1.1475	−0.7128
10	ANAENTIT	AE	1.0552	−0.7489	−0.3423
11	LANSUPPT	LS	−3.0859	0.8114	0.0448
12	SCRNRPTD	SR	0.6618	0.4300	−0.8302
13	IMPTEXPT	IE	−1.6255	−0.1311	−0.7095
14	COMMUNIC	CO	−1.2257	−0.9110	−0.3239
15	STANDARD	ST	−0.3034	−0.0979	−1.2178
16	AN-ARPTWR	AR	0.5958	1.0358	−0.3850
17	PCMAINFR	PC	−2.6431	−0.7581	0.9147

map, we chose to label all three of the axes. However, our interpretation of the third axis was more tenuous then the other two. Included in our analysis for the interpretation of the axes was the dominance rankings of the CASE functions, the clusters closest to each axis, and the strength of a particular function relative to each axis.

Our labeled interpretation of the x-axis is "Most Productivity—Least Productivity" because it roughly correlates with the preference rankings shown in Table II. There seems to be general agreement between the respondents that the functions closest to the right side of Figure 2 afford them much more in terms of productivity than those functions located to the far left side of the figure. At least two inferences can be drawn from this interpretation. First, the respondents may find more productivity improvement with the functions on the far right, gradually reducing their productivity improvement as they make use of functions on the left side. Second, it is possible that some of the respondents do not make use of the functions on the left side of the figure, and they may intuitively feel that these functions would deliver less productivity improvement compared to the ones on the right. It clearly appears from the responses, which are mapped into Table II and Figure 2, that the Data Flow Diagram and the Data Dictionary functions of the CASE product are contributing the most to the software engineer's productivity improvements over manual methods.

The interpretation label given to the y-axis is "Simple Modeling Functions—Complex Modeling Functions." Simple and complex are used here with reference to the power associated with each of the CASE product's functional parts. To illustrate, the functions nearer to the top of the plane in the figure such as "Presentation Graphics" and "Screen & Report Design" (all corresponding letters are listed in the figure legend) deliver simple, although very useful, functionality to the software engineer. "Structure Diagrams" and "Structure Charts," whose base of their flag pole is near the center of the plane, are more sophisticated (powerful) modeling functions, and "Data Flow Diagrams" and "Data Dictionary," along with its associated "Analysis—Entity," could represent maximum modeling power for this group of software engineers.

The z-axis, labeled "Data Modeling—Data Flow Modeling," roughly correlates with the two prevailing schools of thought for modeling an information system. To illustrate, "Data Flow Diagrams" and "Data Dictionary," two functions below the plane in Figure 2, are representative of the Data Flow Modeling methodology while "Entity/Relationship Model" and "Data Model," whose flag pole tops are closest to the top of the page in the figure, are representative of the Data Modeling methodology. As mentioned earlier, we had some difficulty labeling this axis as the remaining functions may be used regardless of which methodology is being used.

In addition to our interpretation of the axes in Figure 2, some inferences and interpretations can be made from the evaluation of the clusters that represent groupings of stimuli based on distance between stimuli. Figure 3 presents the dendrogram of the hierarchical clustering (please refer to the legend in Figure 2). As MDS is almost always used as a descriptive model for representing and understanding the data, we preferred five clusters (visually shown in Figure 2) based on the response data for this work. The SPSS-X CLUSTER program presented all cluster combinations. However, we saw little value in considering more than six clusters or less than four clusters as either of these two groups would have provided less meaningful cluster information. The cluster containing "Data Flow Diagram," "Data Dictionary," and "Analysis—Entity" appears to be a cluster that affects software engineers' productivity in a very similar (and positive) way. In Figure 2, these three stimuli are positioned close to the axis labeled "Most Productivity." This cluster is also located close to the axis labeled "Complex Modeling Functions" which means that a high degree of power is afforded to the user by these stimuli. The combination of "most productive" and "complex modeling" could be interpreted as being beneficial to software engineers' productivity. This cluster also contains the two most common Data Flow modeling tools—the Data Flow Diagram and the Data Dictionary.

The cluster containing "Lan Support" and "PC to MainFrame" appears to affect software engineers' productivity in a similar, but converse, way. The effect on their productivity improvement over manual methods when using these functions appears to be minimal. A review of Figure 2 shows these two stimuli very near the "Least Productivity" axis within the figure. A third cluster containing "Screen-Report," "Present Graph," and others also appears to have similar effects on their productivity. The effect appears to afford the user simpler (less powerful) modeling capability yet contributes positively to productivity. This interpretation is made based on the observation that all of the functions in this cluster, with the exception of one, lay in the upper right quadrant of the plane, and the one stimuli that is not in that quadrant is very close to it.

The fourth cluster in Figure 2 groups the "Communication," "Enterprise Adherence to Standards," and "Im-

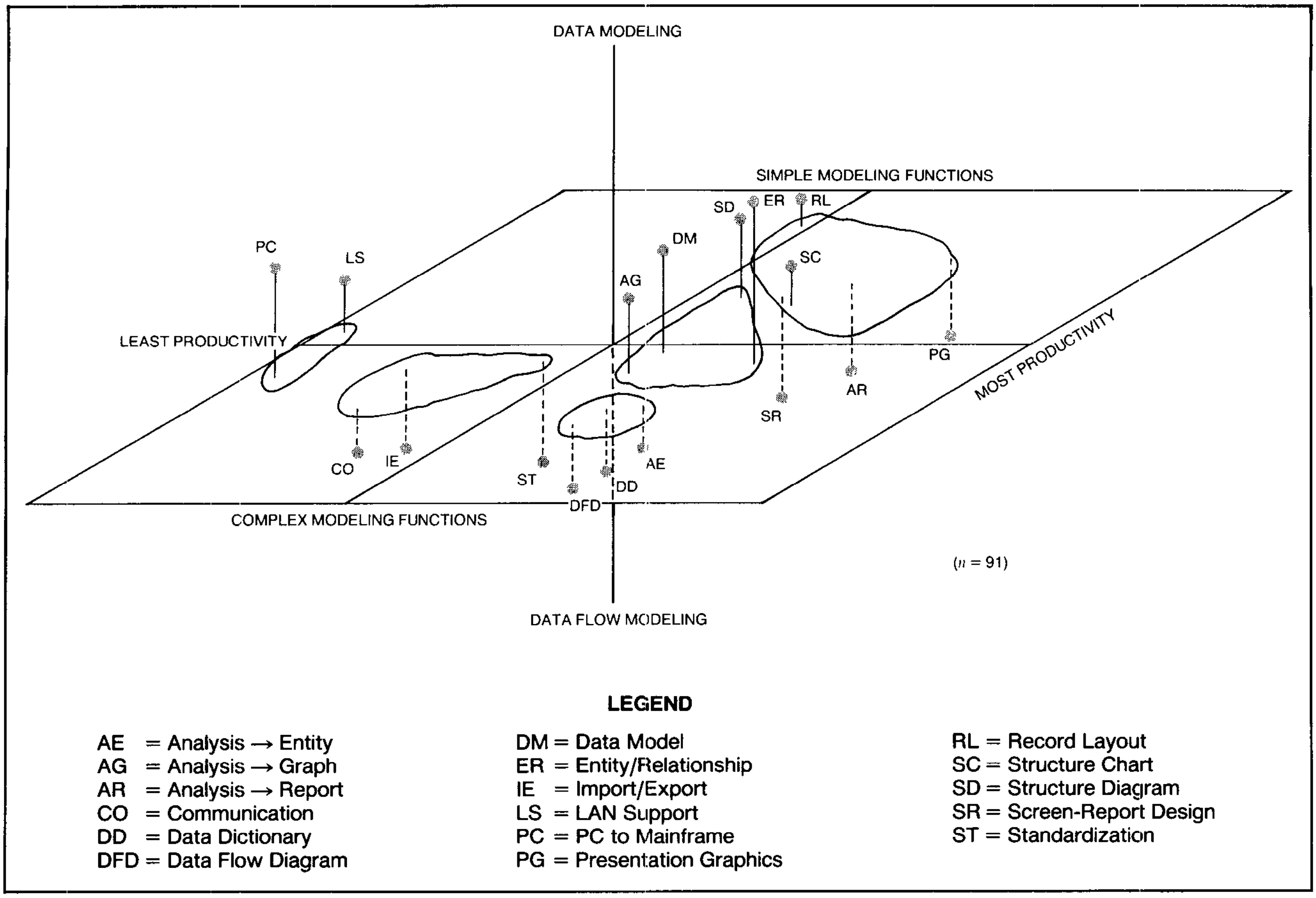

FIGURE 2. Productivity Space for Software Engineers

port/Export Facility" stimuli together. Our interpretation of this cluster is that the three stimuli appear to provide less productivity improvement for the user than the twelve other stimuli. The fifth and final cluster in Figure 2 has grouped the two most prominent Data Modeling tools, Entity/Relationship Diagrams and Data Modeling, with two other stimuli, and the interpretation of this cluster is that these stimuli appear to positively affect productivity over manual methods.

RECOMMENDATIONS

The above observations form the basis for several informational recommendations.

(1) This study shows, via software engineers' perceptions, that their productivity improved with the use of CASE technology.

(2) It identified the functional parts of a specific CASE product that were perceived to provide the most productivity as well as those that offered the least improved productivity. Software engineers that use other CASE products may be able to draw some inferences from this study as many of the CASE products have generic equivalents of the stimuli used in this survey.

(3) The study indicates that there are perceived productivity improvements attributed to adherence to the enterprise's systems development standards when using CASE technology. This is significant since most of the larger enterprises must enforce rigorous system development methodologies and associated standards.

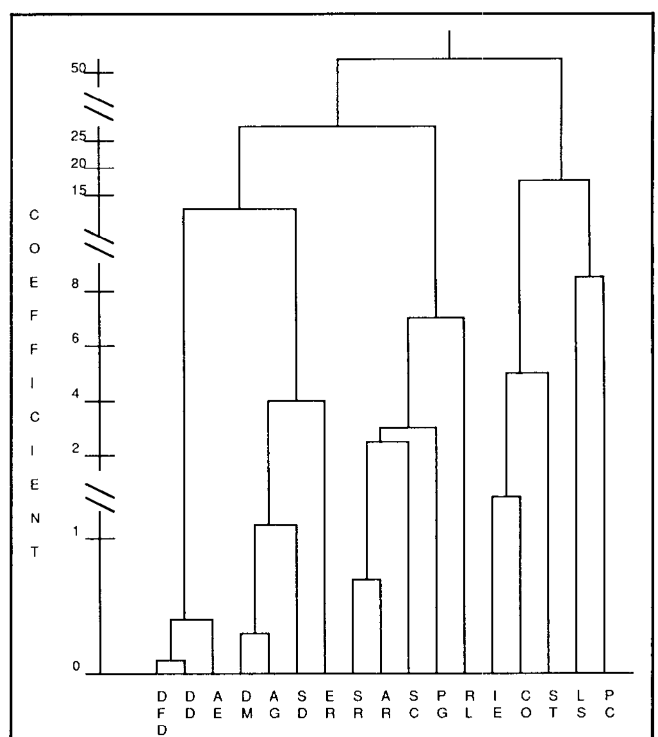

FIGURE 3. Clustering Dendrogram using Ward's Method

(4) This study is a step towards rigorous validation of the effects of CASE technology on software engineers' productivity.

(5) The results imply that CASE technology is per-

ceived to improve productivity, and that adherence to enterprise system development standards deserves enterprise attention and evaluation. Merlyn [11] suggests that CASE technology come under an organizational umbrella called the Development Center. This organization would view the systems development process in the same way as development of an information system. Its charter would be to assist in the investigation, introduction, monitoring, and on-going support of new technologies which can be applied to the systems development process.

(6) Software engineering vendors should continue to enhance their product offerings to address those facets of the software engineer's job that will deliver the greatest increases in productivity coupled with increased system quality.

(7) This study implies that CASE product offerings need not be as robust in functionality as the one used in the study in order to positively affect productivity.

(8) Software engineering researchers should continue to push the frontiers of technology that continue to automate more of the software engineer's job responsibilities. Increased system quality as well as increased productivity should result from this.

SUMMARY AND CONCLUSIONS

This study did not attempt to measure the degree of productivity improvement using CASE technology, but represents a step in that direction. It also did not investigate the many costs associated with implementing the technology, all of which may have an impact on overall productivity. The intuitive claims about improved productivity made by CASE vendors have been supported through this research, while the amount of productivity improvement was not investigated.

As we turn the corner into the 1990s, the information systems being developed are much more sophisticated, integrated, interactive, and distributed than their predecessors. For the human mind to be fully knowledgeable of the system, and do completeness, consistency, and integrity checking in a timely manner, automated support will be required. The software engineering discipline may be moving into an era where large systems development efforts must be assisted by CASE or CASE-like technology.

APPENDIX A

LIST OF STIMULUS ITEM DEFINITIONS

Stimulus*	Definition
1	Graph Analysis helps you verify the design of a project by producing reports on your graphs.
2	Entity List lets you create or modify lists of entities that are used to analyze the contents of the project dictionary.
3	Report Writer lets you produce customized reports on the project dictionary.
4	The advantage is to have [CASE product] not only operational on a PC but also operational on a mainframe computer.
5	The central repository for all definitions and data, and also the clearinghouse for all of the information that is associated with a given project.
6	Representation of the flow of data through a system showing the external entities that are sources or destinations of data, the processes that transform data, and the places where data is stored.
7	A top-down technique that illustrates the data model using data and relationship objects that are connected together.
8	The ability to export information to or import information from another PC or host (mainframe) computer that may have the same or a different CASE tool operating on it.
9	The ability for [CASE product] to be supported on Local Area Networks.
10	A graphical representation of data entities, illustrated by irregular ovals, and the relationships among them, illustrated by connections. The conventions used for the connections generally follow Bachman methodology.
11	A graph type that is used primarily for overview presentations. It features a variety of objects and drawing commands.
12	The [CASE product] plays a role in the communication process between all team members.
13	The [CASE product] enhances an organization's efforts to enforce project standardization.
14	The ability to generate program language source code record layouts for record definitions in the project data dictionary.
15	The facility that lets you create or modify screens and/or reports that may become part of the information system being analyzed and designed.
16	Representation of the modular hierarchy within a system. This graph uses decision diamonds to show the location of function objects, data and control flow symbols to show communication between functions, and loop symbols to show repetition.
17	Representations of hierarchical logic flow using Jackson Structured Programming (JSP) symbols. Separate indicators for sequence, selection, and iteration logic are supported.

* Stimulus List

1. Analysis → Graph Analysis
2. Analysis → Entity List
3. Analysis → Report Writer
4. [CASE product] works on both PC and mainframe
5. Data Dictionary
6. Data Flow Diagram (Gene & Sarson, Yourdon)
7. Entity/relationship data model (Chen or Merise)
8. Import and/or Export Facility
9. LAN Support
10. Logical Data Model diagram (IBM)
11. Presentation Graphics
12. Project member's communication via [CASE product]
13. Project standardization
14. Record Layout Generation
15. Screen/Report Design
16. Structure Charts (Constantine)
17. Structure Diagrams (Jackson)

REFERENCES

1. Everitt, B. *Cluster Analysis*. Halsted Press, New York, 1974.
2. Ferguson, G. A. *Statistical Analysis in Psychology and Education*. 3d ed. McGraw-Hill, New York, 1971.
3. Guttman, L. The structure of interrelations among intelligence tests. In *Handbook of Multivariate Experimental Psychology*, R. B. Cattell Ed. Rand McNally, Chicago, 1965, pp. 438–458.
4. Hanson, S. J., and Rosinski, R. R. Programmer perceptions of productivity and programming tools. *Commun. ACM 28*, 2 (Feb. 1985), 180–189.
5. Hoffnagle, G. F., and Beregi, W. E. Automating the software development process. *IBM Syst. J. 24*, 2 (1985).
6. Jones, T. C. Measuring programming quality and productivity. *IBM Syst. J. 17*, 1 (1978), 39–63.
7. Kendall, M. G. *Rank Correlation Methods*. 3d ed. Charles Griffin & Company Limited, London, 1962.
8. Konsynski, B. R. Advances in information system design. *J. Manage. Inf. Syst. 1*, 3 (Winter 1984/85), 5–32.
9. Konsynski, B. R., et al. PLEXSYS-84: An integrated development environment for information systems. *J. Manage. Inf. Syst. 1*, 3 (Winter 1984/85), 64–104.
10. Kruskal, J. B., and Wish, M. *Multidimensional Scaling*. Sage Publications, Beverly Hills, Ca., 1978.
11. Merlyn, V. The backlog stops here. *Computerworld*. June 22, 1987, 61–66.
12. Newman, P. S. Towards an integrated development environment. *IBM Syst. J. 21*, 1 (1982), 81–107.
13. Norman, R. J. Integrated development environments in support of information systems design methodologies and systems analysts' productivity. Ph.D. dissertation, Univ. of Arizona, 1987.
14. Nowaczyk, R. H. The relationship of problem-solving ability and course performance among novice programmers. *Int. J. Man-Machine Stud. 21*, (1984), 149–160.
15. Nunamaker, J. F., Jr. A methodology for the design and optimization of information processing systems. In *Proceedings of the Spring Joint Computer Conference (SJCC)* (Atlantic City, N. J., May 18–20). AFIPS, Montvale, N. J., 1971, pp. 283–294.
16. Schiffman, S. S., Reynolds, M. L., and Young, F. W. *Introduction to Multidimensional Scaling*. Academic Press, New York, 1981.
17. Shepard, R. N. Multidimensional scaling, tree-fitting, and clustering. *Sci. 210*, 24 (Oct. 1980), 390–398.
18. Teichroew, D. Problem statement analysis: Requirements of the problem statement analyzer (PSA). In *Systems Analysis Techniques*, J. D. Couger and R. Knapp, Eds. John Wiley & Sons, New York, 1974.
19. Teichroew D., and Hershey E. A., III PSL/PSA: A computer-aided technique for structured documentation and analysis of information processing systems. *IEEE Trans. Softw. Eng. SE-3*, 1 (Jan. 1977), 41–48.
20. Thadhani, A. J. Factors affecting programmer productivity during application development. *IBM Syst. J. 23*, 1 (1984), 19–35.
21. Torgerson, W. S. *Psychometrika, 17*, 401 (1952).
22. Torgerson, W. S. *Theory and Methods of Scaling*. John Wiley & Sons, New York, 1958.
23. Veldman, D. J. *Fortran Programming for the Behavioral Sciences*. Holt, Rinehart & Winston, New York, 1967.
24. Ward, J. H. Hierarchical grouping to optimize an objective function. *J. Am. Stat. Assoc. 58*, (1963), 236–244.
25. Wasserman, A. I., and Gutz, S. The future of programming. *Commun. ACM 25*, 3 (Mar. 1982), 196–206.
26. Wiedenbeck, S. Novice/expert differences in programming skills. *Int. J. Man-Machine Stud. 23*, (1985), 383–390.
27. Yourdon, E. T. What ever happened to structured analysis? *DATAMATION* June 1, 1986, 133–138.

CR Categories and Subject Descriptors: D.2.0 [**Software Engineering**]: General—*standards*; D.2.1 [**Software Engineering**]: Requirements/Specifications—*methodologies; tools*; D.2.8 [**Software Engineering**]: Metrics—*software science*; D.2.9 [**Software Engineering**]: Management—*productivity*; H.1.2 [**Models and Principles**]: User/Machine Systems—*human information processing*; J.m [**Computer Applications**]: Miscellaneous; K.6.1 [**Management of Computing and Information Systems**]: Project and People Management—*systems analysis and design; systems development*; K.6.3 [**Management of Computing and Information Systems**]: Software Management—*software development; software maintenance*

General Terms: Experimentation, Human Factors, Management, Measurement, Standardization

Additional Key Words and Phrases: Cluster analysis, Computer-Aided Software Engineering (CASE), multidimensional scaling, preference, psychological models

ABOUT THE AUTHORS:

RONALD J. NORMAN is currently an associate professor of Information and Decision Systems at San Diego State University. He is a holder of the Certificate in Data Processing (CDP). His research interests are computer-aided software engineering (CASE) technology, technology transfer and organizational change issues, and seamless integration of enterprise applications. He has served as facilities chairman for the International Conference on Information Systems and was the program committee chairman for CASE '88. Author's Present Address: Information and Decision Systems, College of Business Administration, San Diego State University, San Diego, CA 92182–0127.

JAY F. NUNAMAKER, JR., is head of the Department of Management Information Systems and is a professor of Management Information Systems (MIS) and Computer Science at the University of Arizona. He is also chair of the Association for Computing Machinery's (ACM) Curriculum Committee on Information Systems. Author's Present Address: Department of Management Information Systems, College of Business and Public Administration, The University of Arizona, Tucson, AZ 85721.