Analysis of Prototyping

INPUT®

ANALYSIS OF PROTOTYPING

ANALYSIS OF PROTOTYPING

ABSTRACT

This report analyzes current self-contained DBMS simulation and prototyping tool products and compares the application development productivity increases gained to those of life cycle development efforts with third generation languages. The report is based on telephone interviews with users who have been actively employing prototyping in their development efforts.

Although no current product in the market satisfies all the prototyping needs of IS departments, all were considered to be at least satisfactory.

The report also addresses future prototyping tool requirements and offers a sample checklist of 23 features that should be assessed when making product comparisons.

This report contains 79 pages, including 24 exhibits.

SIMULATION AND PROTOTYPING TOOLS

CONTENTS

EXHIBITS

I INTRODUCTION

A. PURPOSE

- As part of INPUT's Information System Program (ISP), this report analyzes the growing use of applications simulation and prototyping tools. This research was conducted based on client interest and need to be more productive. Systems development productivity is a critical objective for most IS organizations, and its achievement constitutes a major contribution to cost containment and the organization's ability to reduce its applications backlog.

B. SCOPE

- A selected group of "showcase" prototyping development activities were chosen to focus on advanced prototyping approaches in use. Through this approach the most valuable lessons are documented and can be used to improve clients' application development processes.

- This report highlights the most productive prototyping efforts in use as judged by the vendors sampled. Each vendor was asked to recommend its best customers for INPUT to interview. The report focuses on a mature user cross section that has been using simulating and prototyping tools to constrain costs and reduce applications backlogs. The following IS management personnel will find the report valuable:

- Directors.
- Systems development personnel.
- Programming personnel.
- Client (user) liaisons.
- Information center personnel.

- This report addresses the following questions:
  - How are the respondents using prototyping tools to develop applications?
  - How are simulation and prototyping defined? What product features are necessary?
  - Where have the respondents used prototyping and on what kinds of systems development projects?
  - When in the project development life cycle is prototyping being applied?
  - Why has prototyping been beneficial and what are its weaknesses?

C. METHODOLOGY

- Thirteen vendors were asked to supply customers who they thought were using their product extensively to simulate business requirements and prototype

applications. Exhibit I-1 contains a complete list of products used by respondents.

- Primary research for this report was based on user and vendor responses to he questionnaire found in Appendix B. Users responded to all questions while vendors completed numbers 26-47 of the questionnaire based on the features their products provided. Questionnaire probes were designed to determine the full use of advertised vendor features and productivity claims. The research reflects the progress made in the last 10 years and compares third generation language (3GL) system development productivity results to fourth generation languages (4GL) in use today.

D. RELATED INPUT REPORTS

- Fourth-Generation Languages Update: Potential Unrealized (1985) investigates whether the potential of FGLs is being realized. Mainframe-, mini-, and micro-based products are analyzed as well as IS versus end-user applications. Actual benefits are described and analyses of successes and failures of FGL applications made. The report concludes with a forecast of future FGL products and an analysis of how FGLs will fit in with forthcoming artificial intelligence products.

- Artificial Intelligence (AI) and Expert Systems (1985) predicts the breakout of these tools from academia to business. The report analyzes the strengths and weaknesses of these systems. The strategic potential of AI and expert systems as a competitive weapon is examined. The report also describes the technological, personnel, and organizational requirements for these systems to become useful and beneficial.

- Application Software Development Tools (1985) analyzes software productivity tools, including resource centers, applications generators, and

EXHIBIT I-1

PROTOTYPE TOOLS REPORTED IN USE

TOOL	VENDOR
*Corvet	Analysts International Corp.
Proxy	Data General Corp.
1-2-3	Lotus Development Corp.
*Express E3000	Information Resources, Inc.
*System 38 Utilities	International Business Machines Corp.
dBase III Command Language	Ashton-Tate, Inc.
*IFPS PC & Mainframe	Execucom Systems Corp.
FOCUS, Developed In-House	Automobile Manufacturer
*MANTIS	Cincom Systems, Inc.
*FOCUS	Information Builders, Inc.
COBOL	International Business Machines Corp.
*Mark V	Informatics General Corp.
*UFO	Martin Marietta Data Systems
CICS Command Level	International Business Machines Corp.
*APPGEN	Software Express, Inc.
*System	Comshare, Inc.
*FCS	THORN-EMI Computer Software Inc.
*Natural	Software AG of North America, Inc.
*ADS/ON-LINE	Cullinet Software, Inc.
DLM	Cullinet Software, Inc.
Easytrieve Plus	Pansophic Systems, Inc.

*Vendor prototype products (13) used as the sample basis for this report. The remaining tools were also used by the respondents for prototyping.

programmers' tool kits. Products are analyzed and actual benefits described. The report recommends techniques for improving software development productivity and discusses the roles products, training, and organization will have in improving productivity.

- Decision Support Evolution: Data to Knowledge (1985) investigates the purpose of decision support systems. It analyzes the evolution of decision support from management information through expert systems, including defining where decision support stops and decision making begins. The report addresses the limitations of decision support systems and identifies justification techniques for these systems.

- New Opportunities for Software Productivity Improvements (1985) suggests that the conditions may now be right for some major improvements and investigates the importance of management commitment to improving productivity and to investment in new development systems.

II EXECUTIVE SUMMARY

- This Executive Summary is given in presentation format to help the busy reader quickly review key research findings.

- It provides an executive presentation, complete with script, to facilitate group communications.

  - The key points of this entire report are summarized in Exhibits II-1 through II-5.

  - On the left-hand page facing each exhibit is a script explaining that exhibit's content.

A. USER PRODUCTIVITY IMPROVEMENTS

- Productivity improvements are very significant when compared to respondents' 3GL experience.

- Leading the list of productivity gains was the ability to build the first prototype faster. One respondent commented, "Users love it--they can see something every two weeks on a terminal instead of every two years."

- Major productivity gains were reported in most critical applications development areas.

- Productivity gains display increasing percentage improvements over 3GL approaches as follows:

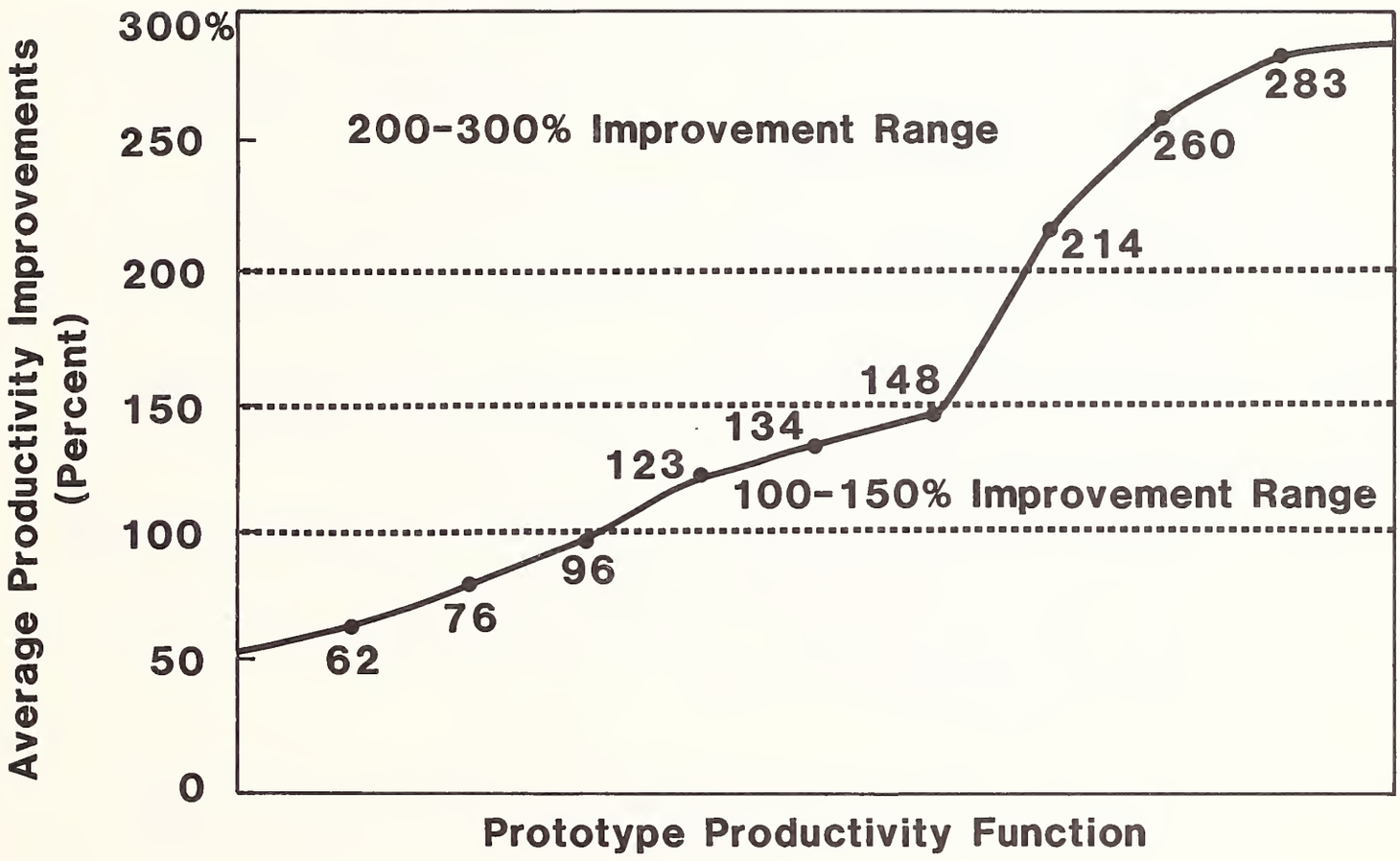
  - A 200-300% improvement range was reported in the time it takes to build the first application prototype along with a reduction in both the number of program statements and the development time devoted to programming mechanics, such as compilations.

  - A 100-150% improvement range resulted from enhanced program code quality and a reduction in training time.

  - A 50-100% range in time savings was due to automatic code generation and data editing. Reduced development effort achieved by sharing program modules which were common to the system represented the minimum savings of 62%.

- The sample user prototyping productivity improvements results are characteristic of a classic productivity function ("S" shaped) with the highest gain resulting in the time it takes to build the first prototype. Exhibit II-1 documents the prototyping productivity function.

USER PRODUCTIVITY IMPROVEMENTS



B. ORGANIZATIONAL IMPACT

- Users forecast a 21% increase in the use of prototyping efforts for each of the next three years. This increment is nearly double the expected IS budget growth.

- Respondents have been applying prototyping for an average of 3.6 years and are very positive about its value. They will continue to use it in the future to reduce the applications backlog.

- Prototyping has been used in a broad range of applications, from producing simple reports (77%) to developing large integrated applications (100%).

- All respondents have applied prototyping in small and large single application systems development projects.

- Prototyping is currently being used for development projects ranging from simple reports to large integrated applications. However, as Exhibit II-2 demonstrates, the proportion of applications being prototyped diminishes in the case of integrated applications.

INPUT®

ORGANIZATIONAL IMPACT

Degree of Complexity

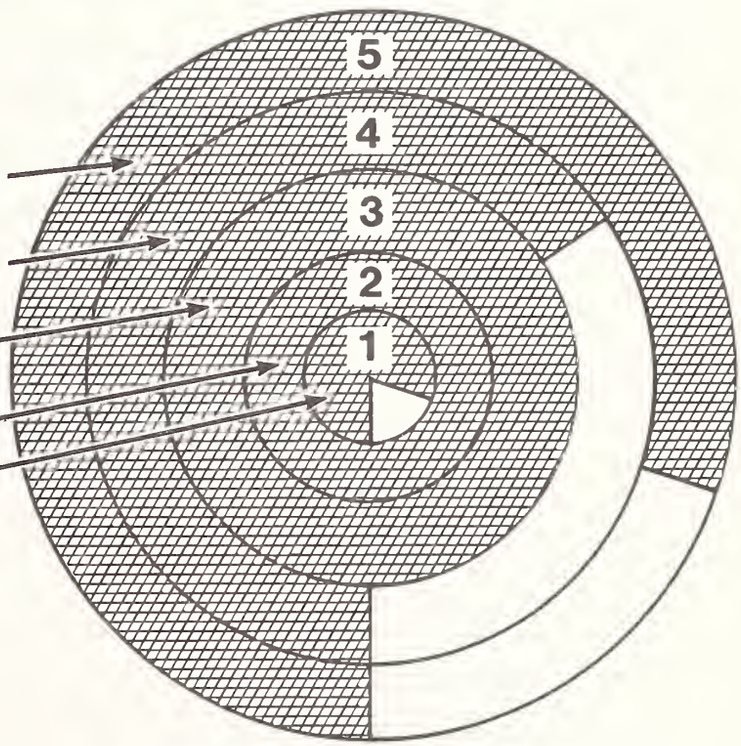
5. Large Integrated Applications

4. Small Integrated Applications

3. Large Single Applications

2. Small Single Applications

1. Simple Reports



Proportion of Respondents' Use of Prototyping by Degree of Complexity

C. LIFE CYCLE IMPACT

- Respondents reported a collapsing of the life cycle development process. Several respondents indicated that the implementation phase was nonexistent or part of the development phase. The new approaches are changing the shape of the old life cycle time distribution curve.

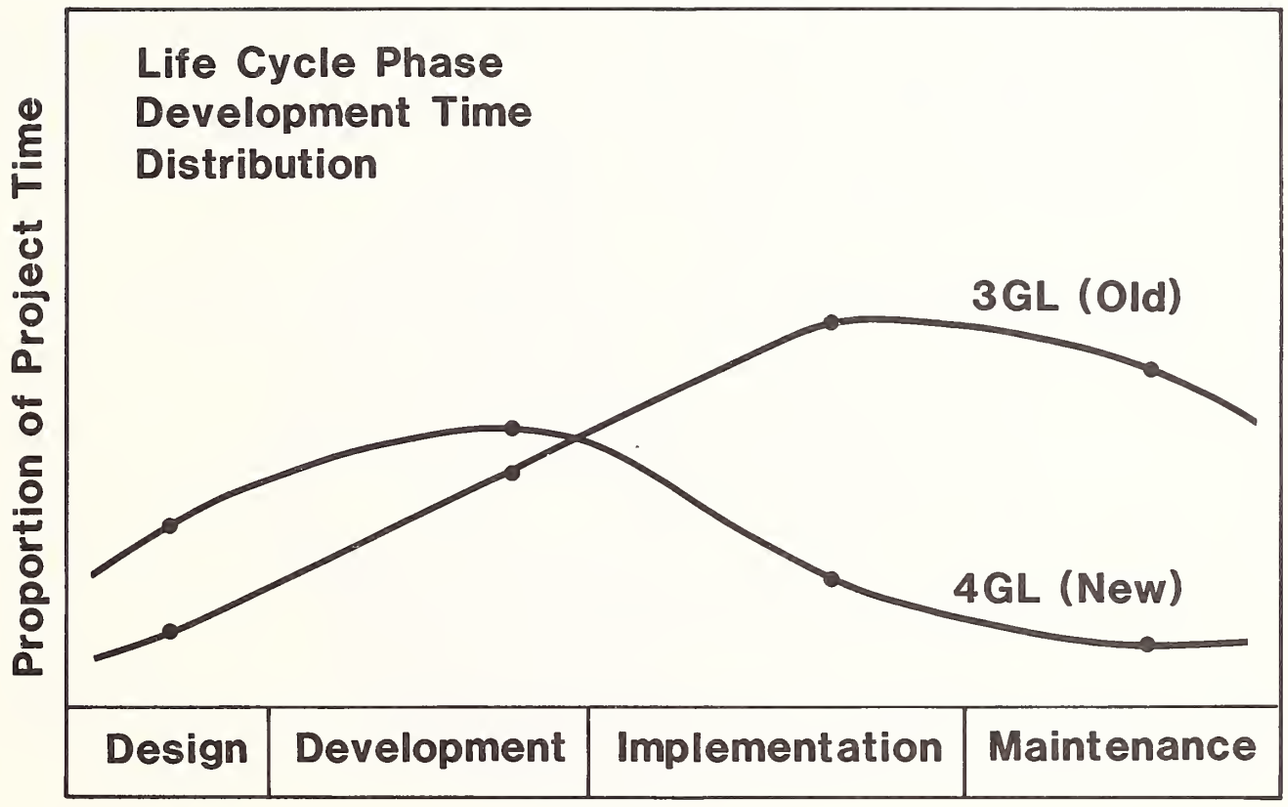
- On average, respondents indicated that they were experiencing a 212% improvement over third generation language development approaches.

- Most of the productivity gains came from the design (217%) and development (243%) phases.

- Better quality systems are the result of user prototyping efforts as evidenced by the ease of making maintenance and updates. This value improved by 118%. Since maintenance often contributes the majority of the life cycle costs, this is potentially very significant.

- Exhibit II-3 represents a conceptual model that describes the distribution of project time for 3GL (old) and 4GL (new). The major difference is the reversal of the proportion of total project time from phases 3 and 4 for 3GL to 1 and 2 for 4GL, again implying lower overall life cycle costs. However, the data does not cover the actual results of an application in full production which may change these initial results.

LIFE CYCLE IMPACT



D. USERS' VENDOR PRODUCT RATINGS

- Most (92%) users rated their tools to be at least satisfactory for prototyping.

- The ratings were evenly split between ratings of 4 for satisfactory and 5 for adequate (on a scale of 1 being inadequate and 5 being adequate). There were no inadequate ratings and only 8% rated the product average. An adequate rating means the vendor's products provide enough facility to meet the organization's existing prototyping needs. An average rating indicated that most prototyping tasks can be performed.

- The major prototyping features desired in order of preference were:

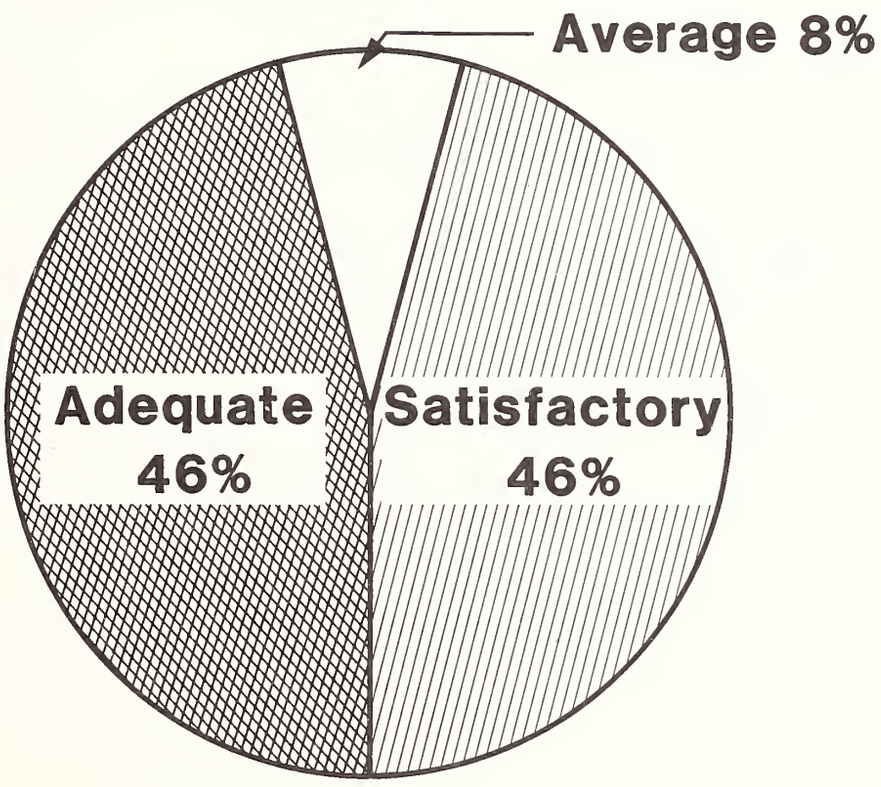
  - Program Development Tool Category.

  - DBMS Support Tool Category.

  - General Support Tool Category.

- Exhibit II-4 shows only the ratings for prototyping, which is one of the major strengths of 4GL technology.

USERS' VENDOR PRODUCT RATINGS



E. I.S. OPPORTUNITIES/FUTURES

- In a word, the respondents were "upbeat." Their prototyping tools have allowed them to:
  - Chip away at the applications backlog.
  - Improve user/IS communications.
  - Demonstrate real success projects for management.
- All respondents are planning extensive use of prototyping for new projects and through other development plans based on productivity gains of the past.
- Users are looking forward to the next generation of prototyping tools. Flexibility, enhanced product speed/capabilities, plus an integrated applications systems requirements definition capability are their expectations in the next three years.
- Exhibit II-5 summarizes the respondents' major plans for prototyping.

INPUT®

INFORMATION SYSTEMS PROTOTYPING OPPORTUNITIES/FUTURES

- Use Prototyping for New Applications
- Most Will Use for More Complicated Projects
- Vendors Need to Enhance Their Existing Products
- There Is a Need for Improved Vendor Products to Keep Pace with Demand

III POPULAR INDUSTRY PROTOTYPE TOOLS

A. BACKGROUND AND DEFINITIONS

- Prototyping as an alternative to the traditional life cycle approach to systems development has been known for a number of years. D. Ness, of the Wharton School, recommended a design approach he called "middle out" in 1975. Other references to early forms of prototyping such as adaptive design and evolutionary design are 10 years old or older. Here we will gauge how far we have come in the past decade.

- Simulation refers to the process of reducing a business requirement to a set of function requirements. Prototyping is the translation of the simulated business requirement into an IS application specification. We will refer to these dual terms throughout this report as prototyping--simulation is always assumed.

- Exhibit III-1 represents a working prototype definition which was validated by the research respondents. The five elements are presented in rank order priority from 1 to 5, based on a scale of 1 being unimportant and 5 being a very important part of the respondent's prototyping methodology.

- The respondents' ratings were consistently high and averaged 4.0 (important). Additional comments were:

EXHIBIT III-1

WORKING PROTOTYPE DEFINITION

RANK ORDER	DEFINITION ELEMENT	RESPONDENTS AVERAGE RATING
First	An iterative learning process is a given for the design phase of a system's development. A model (prototype) is built for user evaluation and is returned to the analyst for updating. This is understood as a heuristic (learning) process for both the system builder and the user.	4.5
Second	To ensure the convergence of user requirements and IS ability to satisfy these needs, users must actively participate in the design through the implementation phase.	4.2
Third	Prototype development depends on relatively fast turn-around or minimum time between iterations. Timely feedback is a prerequisite for effective user and analyst learning.	4.0
Fourth	Integrated hardware and software development tools are required for efficient prototyping.	4.0
Fifth	The initial prototype must be produced at minimum cost. Minimum cost can be difined either as a function of a corporate capital justification policy or based on the organization's past experience. The latter unit of measure is probably the more realistic of the two, since it requires productivity improvement and can be measured over time.	3.3

- "Our major concern is to reduce the number of iterations; we want better communications between users and developers."

- "Ease of use."

- One user suggested that prototyping provided "political support, high level entry" for his systems development efforts.

B. PRODUCT FEATURES FOR PROTOTYPING

- Depending on the level of effort and/or the kind of application project being developed, a wide variety of tools are available. The possibilities run the gamut from analysis packages like Lotus 1-2-3 through application generators such as IBM's ADF. But given the impact of data base management systems (DBMS), this research concentrated on self-contained DBMS languages and DBMS product suites which are actually an outgrowth of DBMS. The three major feature categories used to evaluate these products were as follows:

  - General support features.

  - Data management support features.

  - Program development features.

- Through an examination of existing products, 24 detailed features were developed which specifically fit in the 3 major categories above. A list of these features and their definitions are found in Appendix A. Exhibit III-2 presents the products selected and vendor-defined product features. Company support data for each of the respondent vendors is presented in Appendix C.

EXHIBIT III-2

SELECTED VENDOR PRODUCT FEATURES

FEATURE	ANALYSTS INTERNATIONAL CORPORATION	INFORMATION RESOURCES INCORPORATED	INTERNATIONAL BUSINESS MACHINES	EXECUCOM SYSTEMS CORPORATION	CINCOM SYSTEMS INCORPORATED	THORN-EMI COMPUTER SOFTWARE INCORPORATED	INFORMATION BUILDERS INCORPORATED
General Support Features							
Remote Job Entry	X		X		X	X	X
Teleprocessing Monitor	X		X	X	X	X	X
PC/Mainframe Link	X	X	X	X	X	X	X
Audit (File Fooring, etc.)	X	X	X	X	X		
Security (Data/Resource)		X	X		X	X	X
Optimized Compiler	X		X			X	
Linked Applications Software	X	X		X	X		
High Quality Graphics		X	X	X	X	X	X
OPS System Independent	X	X		X	X	X	X
Data Management Support Features							
Relational Data Base	X	X	X	X	X	X	X
Data Dictionary	X	X					
Distributed Data Base Support	X	X	X		X	X	
Data/Language Independence	X	X	X		X	X	X
Program Development Features							
On-Line Development	X	X	X	X	X	X	X
Batch Development	X	X		X	X	X	X
Interactive Debugging		X	X	X	X	X	
Code Generation/Reuse	X	X		X	X	X	X
Automatic Documentation	X	X		X	X	X	X
Query/Retrieval	X	X	X	X	X	X	X
Screen/Report Writer	X	X	X	X	X	X	X
English Language Query		X	X		X		X
Foreign Language Conversion		X		X	X		
Text Editing/Formatting		X		X	X	X	

Continued

EXHIBIT III-2 (Cont.)

SELECTED VENDOR PRODUCT FEATURES

FEATURE	INFORMATICS GENERAL CORPORATION	MARTIN MARIETTA DATA SYSTEMS	SOFTWARE EXPRESS INCORPORATED	COMSHARE INCORPORATED	CULLINET SOFTWARE INCORPORATED	SOFTWARE AG INCORPORATED
General Support Features						
Remote Job Entry	X	X			X	X
Teleprocessing Monitor	X	X			X	X
PC/Mainframe Link				X	X	X
Audit (File Footing, etc.)			X		X	X
Security (Data/Resource)	X	X	X	X	X	X
Optimized Compiler	X	X	X			X
Linked Applications Software			X		X	X
High Quality Graphics				X	X	X
OPS System Independent	X	X	X	X	X	X
Data Management Support Features						
Relational Data Base			X	X	X	X
Data Dictionary			X		X	X
Distributed Data Base Support	X	X		X	X	X
Data/Language Independence	X		X	X	X	X
Program Development Features						
On-Line Development	X	X	X	X	X	X
Batch Development	X		X		X	X
Interactive Debugging	X	X	X	X	X	X
Code Generation/Reuse	X				X	X
Automatic Documentation	X		X		X	X
Query/Retrieval		X	X	X	X	X
Screen/Report Writer		X	X	X	X	X
English Language Query	X	X	X	X	X	
Foreign Language Conversion			X		X	
Text Editing/Formatting	X	X	X	X	X	X

- The vendor sample used for this research is estimated to be 25-33% of the total self-contained DBMS products market with an above average proportion of DBMS product suites. This sample represents:

  - A total of 10,120 installed packages.

  - An average cost per package of $74,400--ranging from $34,000-127,000.

  - Average total company sales of $82.2 million, supported by a mean of 755 employees.

- Roughly one-half of the vendor products were introduced in the 1972-1979 time period (55%), the rest between 1980-1982 (45%). They represent relatively new product offerings.

- Many vendors have targeted their products to be supported by IBM's large-scale mainframes (with the preponderance being 4300 series and up). Other equipment vendors are, however, liberally represented.

- No sample vendor could boast of providing all 23 of the prototyping features listed in Exhibit III-2. On average, the vendor sample claimed to meet 74% of the feature requirements. The distribution of features supplied by the vendors that responded are as follows:

  - Thirty-one percent supplied 87-96% of the total.

  - Thirty-one percent supplied 70-78% of the total.

  - Thirty-eight percent supplied 48-65% of the total.

- This analysis suggests that the user must have a good specification for the facilities he requires and carefully evaluate vendor offerings based on the kind of facilities required to prototype.

C. SYSTEM DEVELOPMENT LIFE CYCLE IMPACT

- Respondents were asked to estimate the percentage improvement for the entire life cycle from design phase to implementation phase. On average, they estimated that the entire life cycle improvement was 212%. However, it is interesting to note that the major improvements were found in the first two phases, 217% for design and 243% for development.

- Several respondents indicated that the implementation phase was nonexistent or was now part of the development phase. These marginal comments suggest a collapsing of the life cycle process. Given the improvements cited, it is not just a shift in effort—it is a real reduction in the time to deliver the system. Maintenance changes and updates also were reported improved by 118%. This would indicate a lasting productivity impact affecting even the maintenance of the new system.

D. ORGANIZATIONAL IMPACT

- The average annual growth rate for prototyping activities was reported to be 21% which is considerably greater than the respondents' total IS budget growth rates of 12%. These growth rates represent the respondents' projections for the next three years, indicating sizable future increases.

- Respondents were asked to define their applications prototyping experience level. A series of probes was presented which indicated increased complexity from (1 = simple) "uses prototyping to develop single reports" to (5 = complex) "uses prototyping to develop large integrated applications." These results, and those of a question regarding the length of time prototyping had been a formal part of the applications development process, substantiate advanced prototyping implementation. On average, the group has been using prototyping for

3.6 years and has progressed through all five levels of complexity, as shown in Exhibit III-3.

- Experience was heavily concentrated (100%) in the small to large single applications levels (2 and 3) with substantial experience (69% and 77% respectively) in the small and large integrated applications levels (4 and 5). It should be noted that many respondents skipped the first project level (simple projects) and went directly to levels 2 through 5. This analysis clearly shows prototyping is being used in more complex projects.

E. USER PRODUCT FEATURE REQUIREMENTS

- Since this research draws on user experience with 13 prototyping products, the list of features developed represents a prototyping "tool box" in current use.

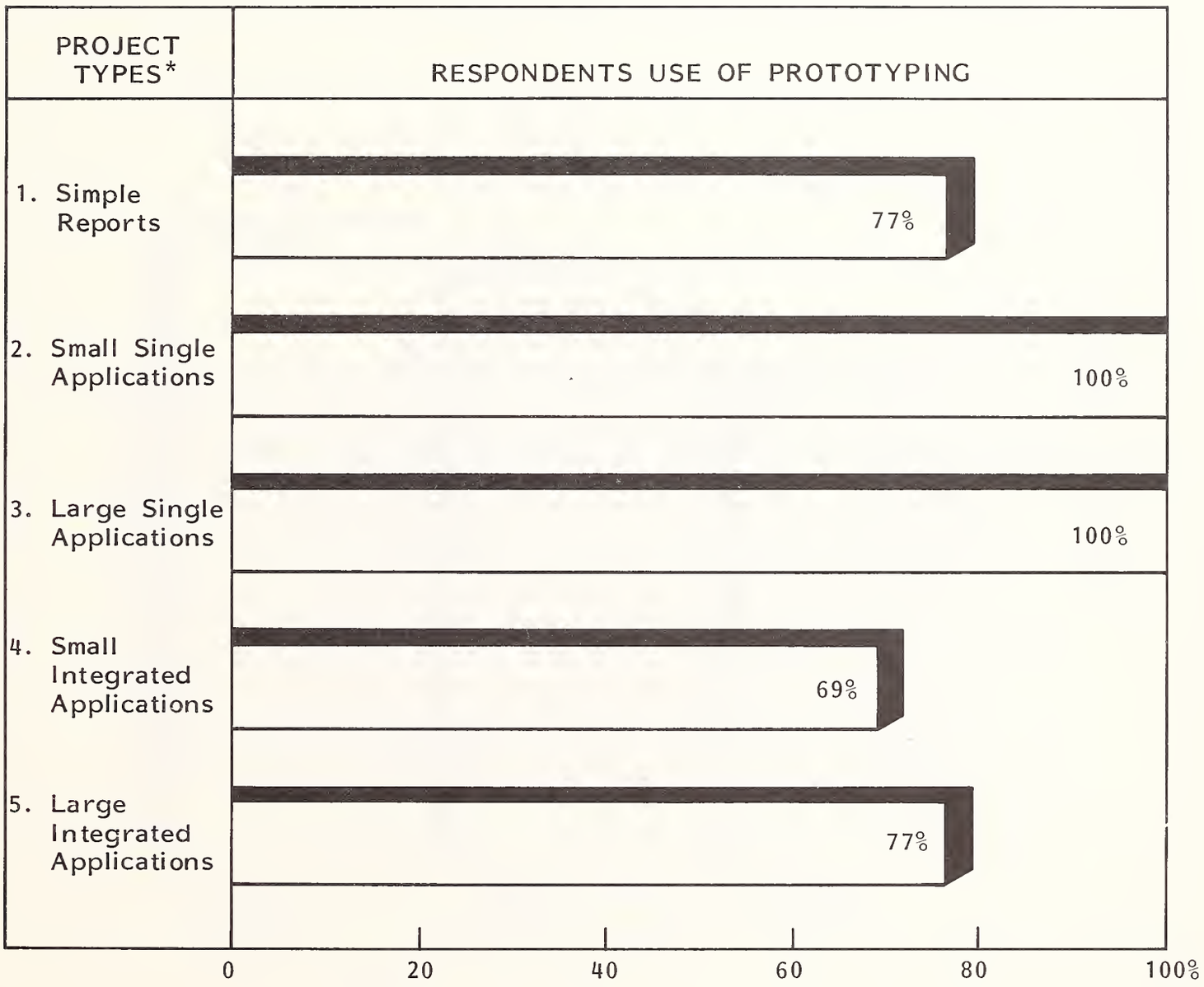
- Respondents were asked to rate the importance of the tool features on a scale of 1-5 (1 being unimportant and 5 being very important). They were also requested to rate, on the same scale, the amount of technical knowledge required to use the features with 1 being no technical IS knowledge required to 5 representing advanced technical IS knowledge required. As might be expected, the three major categories (general support, DBMS support, and program development support) were rated progressively more important.

- Each of the major categories described in Exhibit III-4 are analyzed to isolate the most important integrated hardware/software development tool features. Exhibits III-5 through III-7 present both the importance of the feature and the level of technical IS knowledge required to use it.

- General support features included nine items, five of which were rated above the average value of 3.3. The features which were regarded as of above average importance were security, operating system independence, audit, and PC/mainframe link. Today's emphasis on protecting corporate assets and the

EXHIBIT III-3

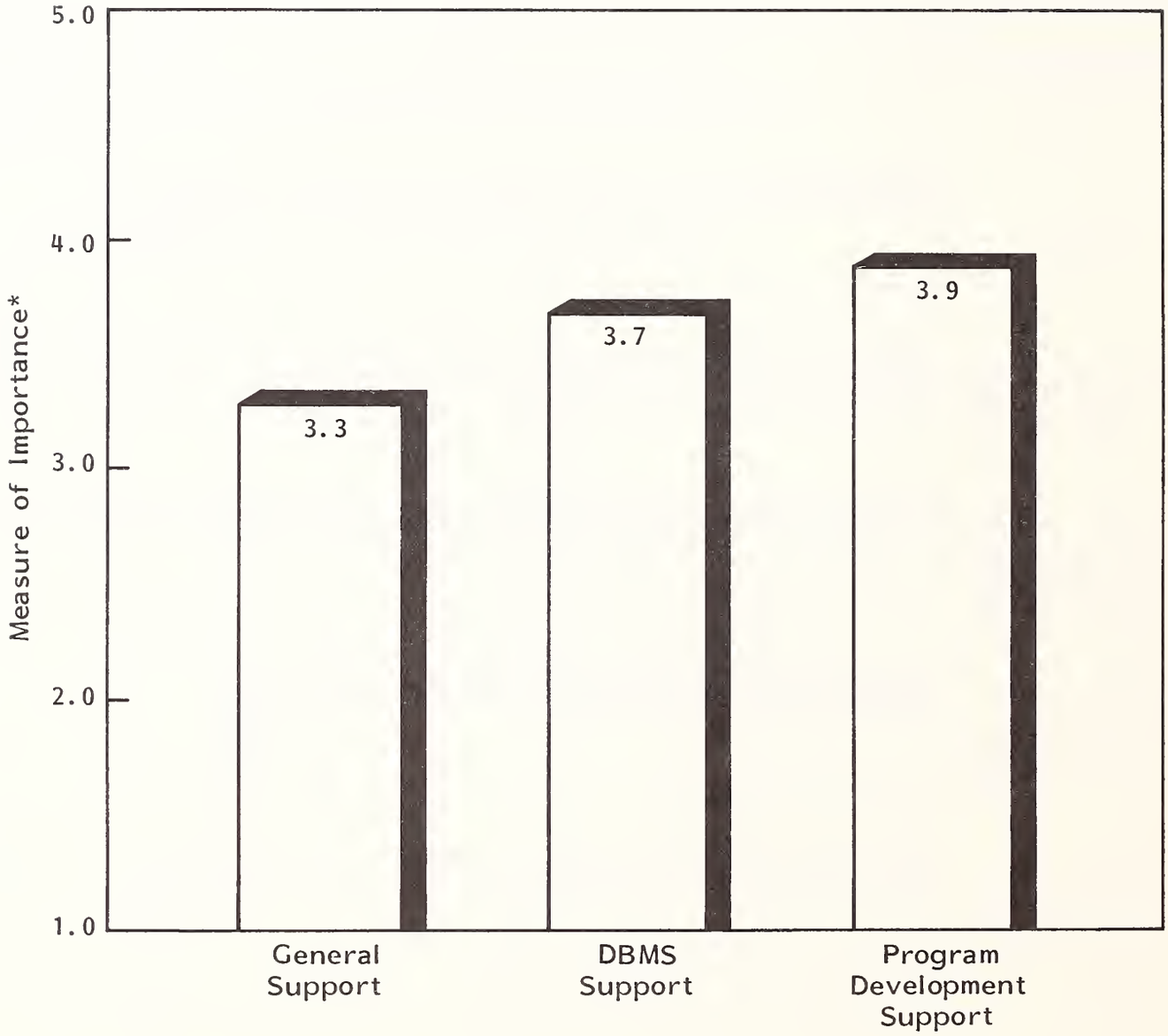
PROTOTYPING EXPERIENCE LEVEL



*In Order of Complexity: 1 = Simple, 5 = Complex.

EXHIBIT III-4

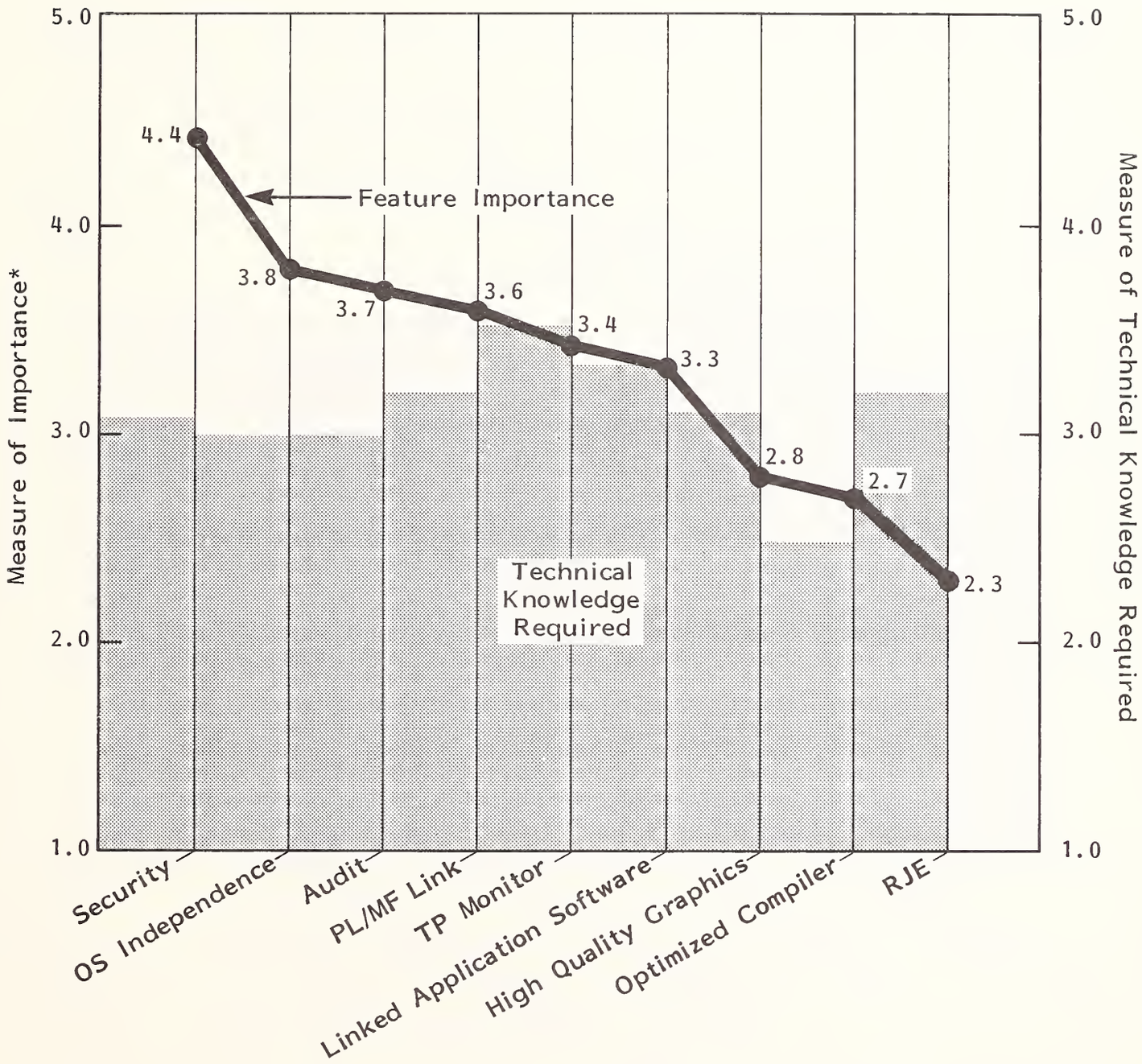
PROTOTYPING SUPPORT TOOL CATEGORY IMPORTANCE



*Rating: 1 = Unimportant, 5 = Very Important

EXHIBIT III-5

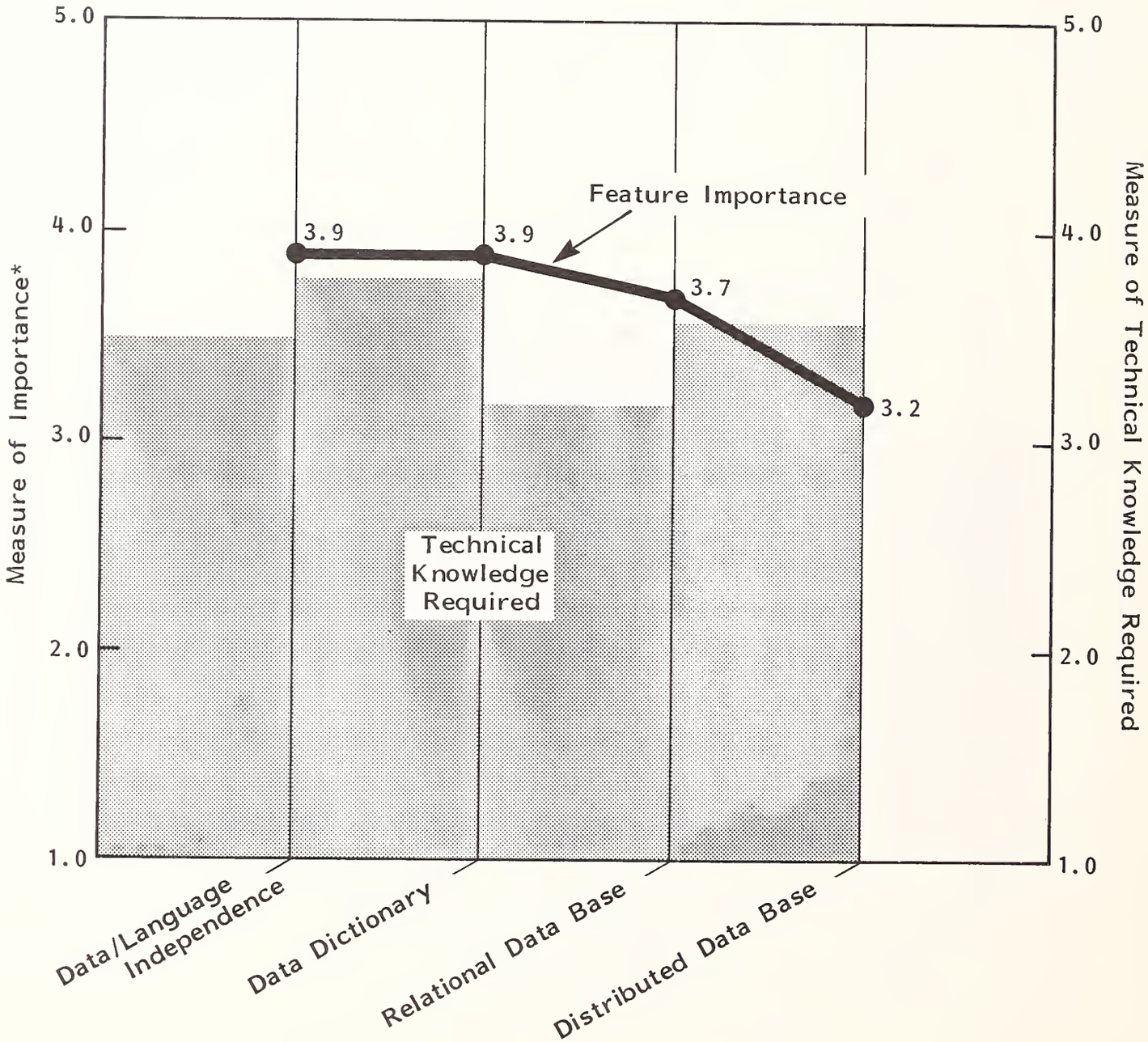
GENERAL SUPPORT FEATURES
IMPORTANCE AND TECHNICAL KNOWLEDGE REQUIRED



*Rating: 1 = Unimportant, 5 = Very Important

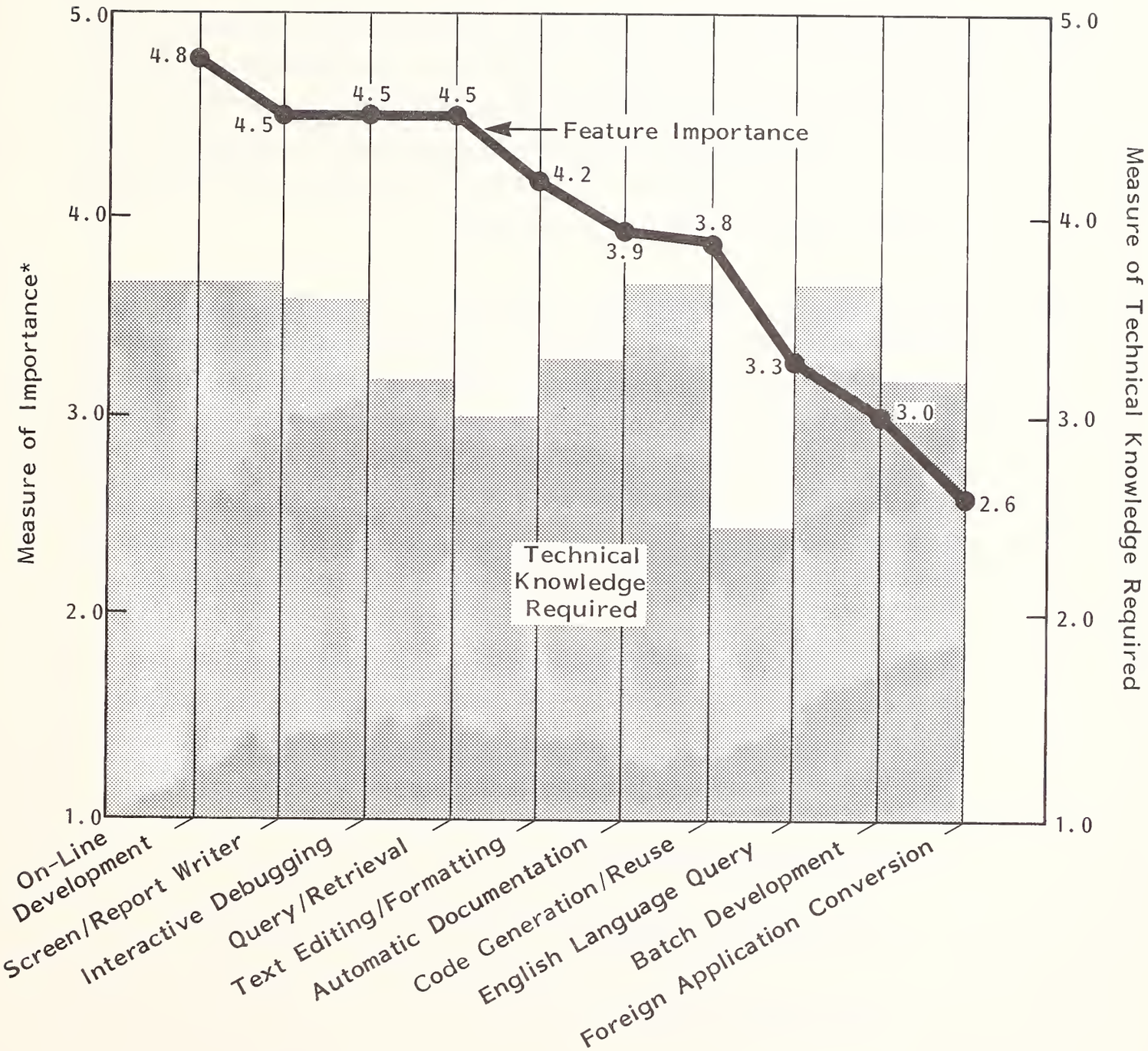
EXHIBIT III-6

DBMS SUPPORT FEATURES
IMPORTANCE AND TECHNICAL KNOWLEDGE REQUIRED



*Rating: 1 = Unimportant, 5 = Very Important

PROGRAM DEVELOPMENT SUPPORT FEATURES
IMPORTANCE AND TECHNICAL KNOWLEDGE REQUIRED



*Rating: 1 = Unimportant, 5 = Very Important

continued effort by the accounting fraternity to have application auditability built in from the beginning make the security and audit features' importance understandable. Operating system independence can be considered a part of most tools, and the fourth-place entry of the PC/mainframe link shows the importance of emerging systems development workstations. Those features which were rated as below average in importance can be obtained outside the product used or as an adjunct to prototyping. For example, one respondent said that he did have an important need for high quality graphics, but he used separate system resources to satisfy this requirement. Exhibit III-5 details the results of these queries. Each feature is presented with its associated technical knowledge requirement evaluation.

- Moving closer to the immediate needs of the prototyper, DBMS support features of the greatest importance were data language independence, data dictionary, and relational data base. It should be noted that no attempt was made to determine what level of sophistication the respondents' relational data base support included. Most agreed that they had relational data base capabilities. The data dictionary level of technical knowledge rates as one of the most demanding features. Using the same metrics as explained above, Exhibit III-6 presents the relative importance of the DBMS support features.

- Of all the prototyping features explored, program development support tools proved to be the most important. Ten items were assessed, and 60% of them were rated as "important" or approaching "very important." On-line development led the list, following five important companion features:

  - Screen/report writer (4.5).
  - Interactive debugging (4.5).
  - Query/retrieval (4.5).
  - Text editing/formatting (4.2).
  - Automatic documentation (3.9).

- Batch processing and foreign application conversion are not important features. The English language query feature which is touted to be for end-user development is also not considered to be of great importance, even though it is reported to be easy to use. Exhibit III-7 summarizes these results.

- Based on the respondent feature preferences, integrated hardware/software development tools can be more definitively documented. The "tool box" referred to at the beginning of this section should have the features or functions listed in Exhibit III-8.

EXHIBIT III-8

MAJOR SOFTWARE TOOL FEATURES DESIRED

- GENERAL SUPPORT TOOL CATEGORY
  - Security
  - Operating System Independence
  - Audit
  - PC/Mainframe Link
- DBMS SUPPORT TOOL CATEGORY
  - Data/Language Independence
  - "Relational" Data Base
  - Data Dictionary
- PROGRAM DEVELOPMENT TOOL CATEGORY
  - On-Line Development
  - Screen/Report Writer
  - Interactive Debugging
  - Query/Retrieval
  - Text Editing/Formatting
  - Automatic Documentation

IV USER PRODUCT ANALYSIS

A. STRENGTHS

- Those surveyed were asked to make a comparison between their third generation language (3GL) experience and the 4GL prototyping tools in use. Nine probes were used to measure order of magnitude productivity improvements that they experienced. The results (presented in descending order of improvements) are:

  - Shorter time to build the first prototype.
  - Reduction in the number of programming statements.
  - Program debugging time improvements related to performing programming mechanics such as compilations, opens, etc.
  - Better reliability and readability of application code.
  - Improvement in program modifications due to fewer syntax errors.
  - Reduction in the training time to be productive.
  - Design improvement due to automatic code generation; e.g., data definitions, screen formatting, processing logic, etc.

- Production systems improvement due to use of automated data editing features.

- Increased ability to share program modules among applications.

- While the range of experience was broad, the individual results tended to cluster about the average. In fact, given the diversity of the respondents' commercial interests, the average values presented in Exhibit IV-1 reflect favorably on increased productivity resulting from the use of these prototyping tools.

- Percentage improvement ranged from a low of 62% to a high of 283%. The respondents' marginal comments are interesting and expand upon the statistical results. While it was evident that those questioned have realized substantial productivity gains, they made the following comments.

  - "Not easy to make syntax errors."

  - "Almost impossible to make syntax errors, big reduction over 3GL."

  - "Final amount of code reduced."

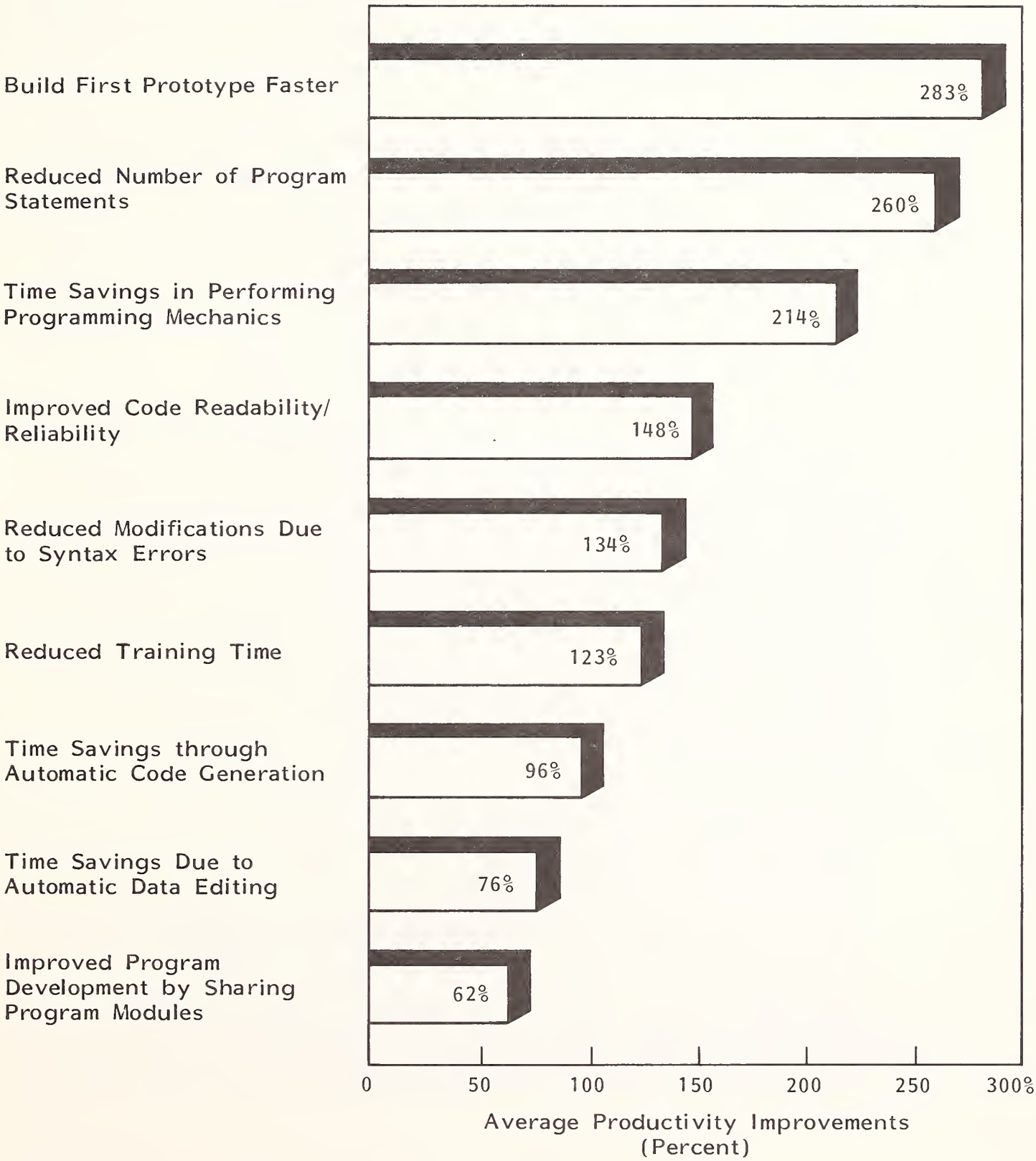
  - "First prototype done in 3 days versus 20 days with our 3GL."

B. WEAKNESSES

- Since prototyping is understood to be an iterative process, where the user and developer are moving jointly toward a final applications system, most weaknesses are seen as deficiencies. However, users do feel that a high degree of technical knowledge is required and that future products (fifth

EXHIBIT IV-1

USER PRODUCTIVITY IMPROVEMENTS



generation languages) should be less technical. Comments regarding future products, specifically referred to as 5GLs, indicate a maturing of the present methodology. Perhaps the fact multiple respondents referred to their existing tools as a methodology and not as a technology as the question was framed indicates that much of what is now done manually or based on product experience shoud be internal to 5GLs.

- This suggest that new products should be designed with an expert system approach. The technical skills required to use today's 4GLs could be automated to allow the developer to spend more time concentrating on productivity and not language mechanics. This thesis was succinctly put by one respondent as follows: "We have a 300% productivity increase, all tools peak out. We are at the peak, satisfied now, but will have to go to a 5GL. . .cannot wear well forever."

- An extensive learning curve was noted by several users, and others indicated no training productivity improvement over 3GLs. After all, these are new languages and take time to learn. Existing products are inflexible and have definite restrictions related primarily to the user not knowing the internal working structure of the tools. Additional training beyond the introductory courses would address this weakness. Excessive system resource usage, especially when large amounts of data are involved, presents some design problems. This requires tradeoffs between adaptability, speed, and productivity requirements.

- A major capability which none of the surveyed products address is an automation/integration of requirements definition specification. This deficiency is demonstrated by comments regarding the amount of technical knowledge needed and the requirement for "programmers to have in-depth knowledge of the application being built." Having an integrated requirements definition specification module would add an additional degree of freedom to the detailed design phase. For example, screens/reports, logical data base, and files specifications could be prespecified and validated by users before the

detailed designing phase. This would allow programmers to concentrate on efficient code generation.

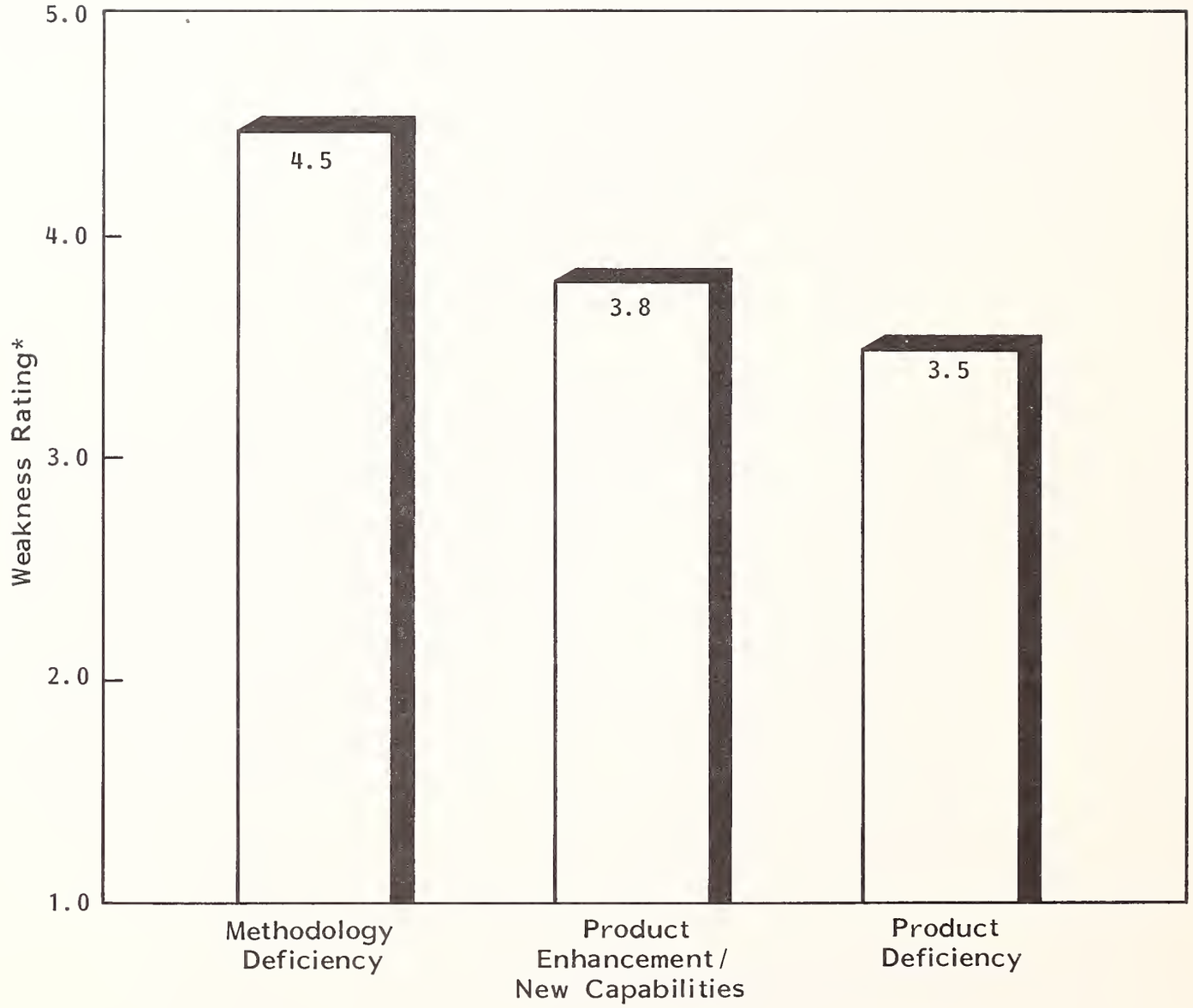
- Other comments which fall into the "wish list" category were:
  - Better/high speed tools.
  - Windowing and split screen capabilities.
  - Improved multi-session facilities.
  - Reduction in unpredictable errors.
  - Offering true report screen painting.
  - Improved documentation capability.
- Overall, the weaknesses do not represent a condemnation of the products. On the contrary, users are in general satisfied, but vendors would do well to document and incorporate their suggestions when designing follow-up products. In fact, the average weakness rating was 3.6 on a scale of 1-5, where 1 was minor and 5 major. User responses were grouped into three areas: methodology deficiencies (e.g., more participation by users); product enhancements/new capabilities (e.g., reduce technical skill required, automate requirements definition phase), and product deficiencies (e.g., improve documentation). The results of this analysis are presented in Exhibit IV-2.

C. I.S. OPPORTUNITIES/FUTURE

- Prototyping has provided a means to chip away at the applications backlog and improve user communications and, in several cases, has demonstrated real

EXHIBIT IV-2

PROTOTYPING WEAKNESSES



*Rating: 1 = Unimportant, 5 = Very Important

success stories for upper management. Perhaps the strongest endorsement for the value of this approach is seen in Exhibit IV-3. A series of questions sought to measure the respondents' propensity for continued use of prototyping tools for developing new and more complicated applications.

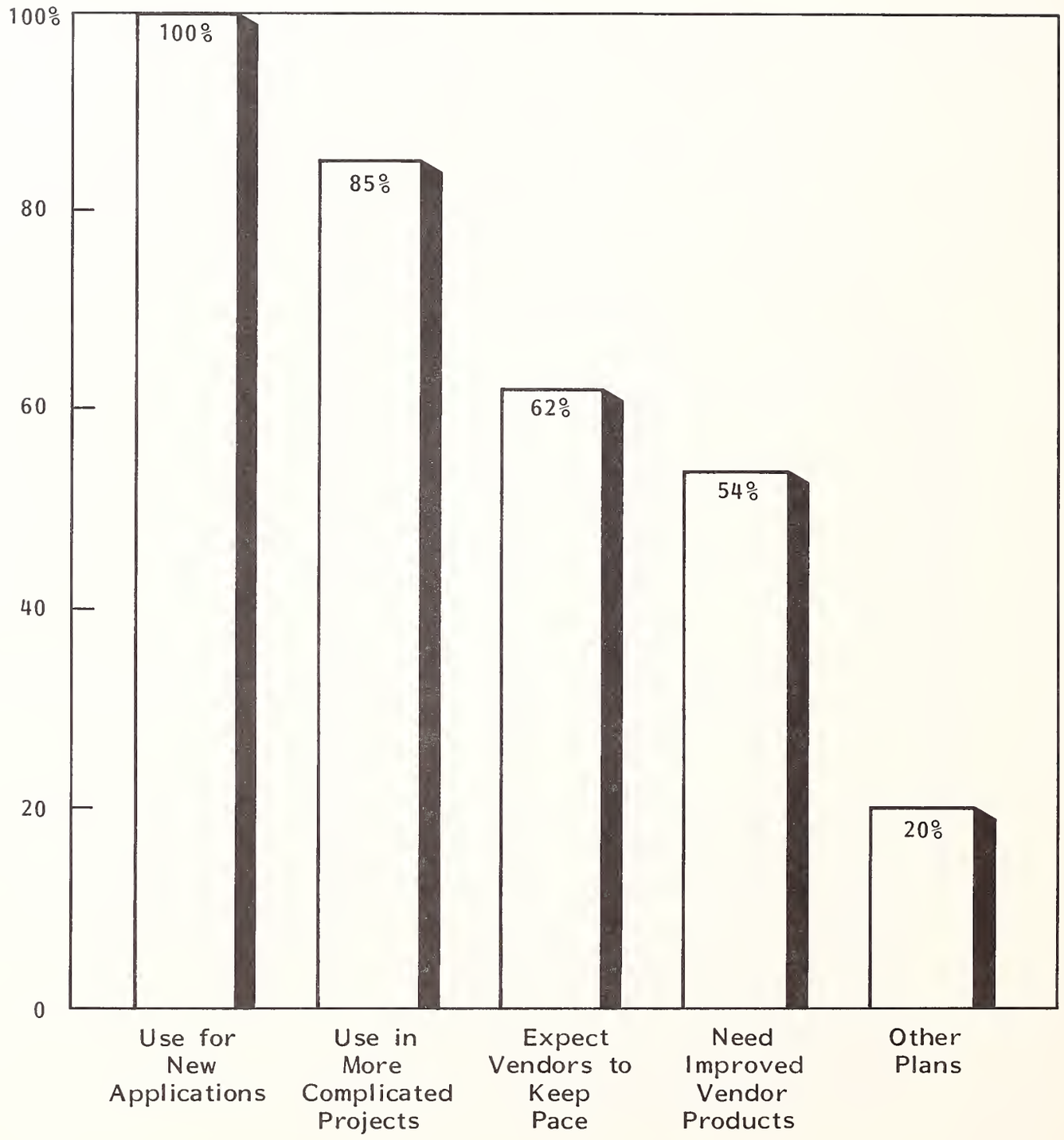
- New applications will be attempted using prototyping for a broad range of reasons, including increased productivity, new department computing requirements, and the continued movement to on-line system development. The improved dialogue between end users and programmers also has contributed to the success of prototyping. Prototyping has given these developers new courage to address the outstanding development requirements of their organizations. The following are representative comments from prototyping supporters.

  - "We use it everywhere."
  - "Whatever comes down the road."
  - "Its value is to improve communications from inception to implementation."
  - "Users love it--they can see something every two weeks on a terminal instead of every two years."

- More complicated projects were not in every respondent's plans. One answered this question in the positive, but said, "if it gets anymore complicated, I'm going to quit!" Responses that were more typical were related to the use of prototyping to stay abreast of the applications backlog, to reduce the timeframe to get projects going, and simply "because it is better." The developers' ability to build on past successes and the increased applications development demand bodes well for increased use of prototyping for even more challenging projects.

EXHIBIT IV-3

I.S. OPPORTUNITIES/FUTURE



- Having tasted success, developers expect vendors to keep pace. They are satisfied with the current vendor support and improved versions, but hope that improvements will continue. Specifically, they are looking for fifth generation languages to include:

  - Improved ability to keep prototyping documentation versions in sync with the code.

  - A link between the requirements definition and detailed design phases.

  - Data base design tuned to flexible applications development prototyping; e.g., data modeling at the logical architectural level.

  - Improved features like windowing, split screens, multi-session capabilities.

  - Improved vendor response time for enhancements that incorporate user suggested improvements.

  - Fifth generation language availability within the next three years.

V FUTURE PROTOTYPING TOOL REQUIREMENTS

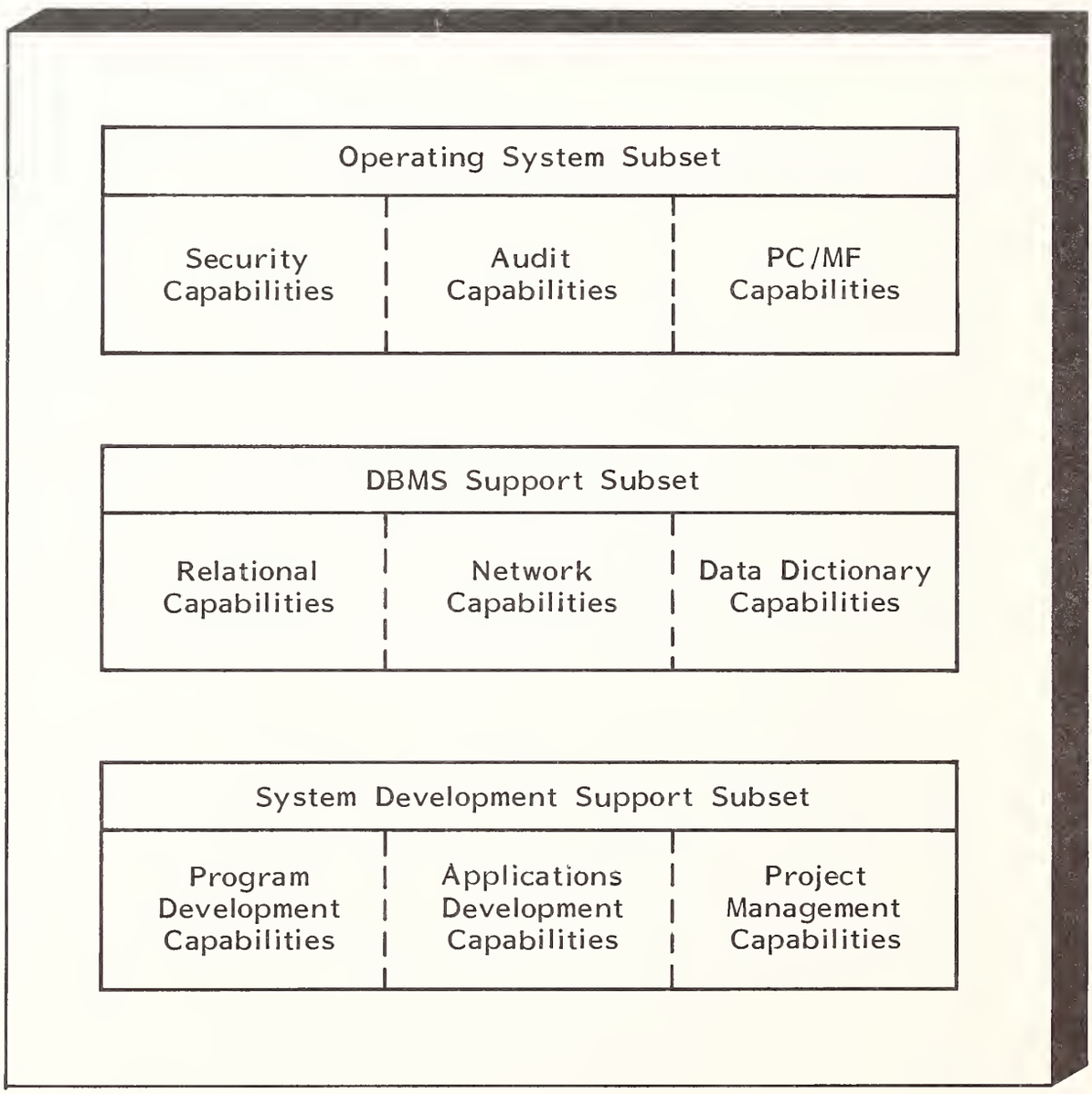
A. PROTOTYPING DEVELOPMENT ARCHITECTURE

- This final section presents a look at future prototype technology. It is a three-year technology extrapolation based on the needs of the respondents, documented product deficiencies, and existing vendor products. For prototyping to mature into a technology from its present methodology status, vendors and users alike need to consolidate their collective experience. User experience to date should be tapped to begin developing the next generation of tools which we will call fifth generation prototyping languages (5GPL). This term was mentioned many times by users, so it has a research basis.

- The prototype development architecture presented includes the most important capabilities documented by this research, plus those capabilities which will make protoyping easier to use, more flexible, and able to contribute to improved productivity in the area of systems development. Exhibit V-1 outlines the three improved subsets of existing product offerings which are required to meet 5GL prototyping requirements. Each of these future product subsets are further detailed in this section. Exhibit V-1 also defines the most important additional capabilities required for the operating (general) system, DBMS, and a major new set of capabilities called the system development support subset.

EXHIBIT V-1

FIFTH GENERATION PROTOTYPING LANGUAGE ARCHITECTURE



B. GENERAL SUPPORT TOOLS

- Links or interfaces to the operating system need to be developed to ensure data integrity, information resource security, application auditability, and personnel computer (PC)/mainframe (MF) connectivity. The PC/MF link will satisfy the need for prototyping workstations. Data and resource security is required to prevent unauthorized data access and unintentional damage or removal of corporate information resources. Either existing standard security packages or integrated protection software are required. Standard audit routines are necessary to allow automated audit functions and must be part of the initial design of the applications system during the prototyping phase. Exhibit V-2 details these features.

C. DBMS SUPPORT TOOLS

- An integrated DBMS, using the most powerful attributes of the evolving relational model, needs to be developed to include networking and data dictionary capabilities.

- Production, special (e.g., departmental systems, etc.), test, and prototype applications require separate, partitioned modules as shown in Exhibit V-3. These would offer more powerful prototyping capabilities for the flexible development and migration of applications from prototype to production status. Facilities such as data manipulation, data base access, data definition, update, and storage would be enhanced.

EXHIBIT V-2

GENERAL SUPPORT TOOLS SUBSET

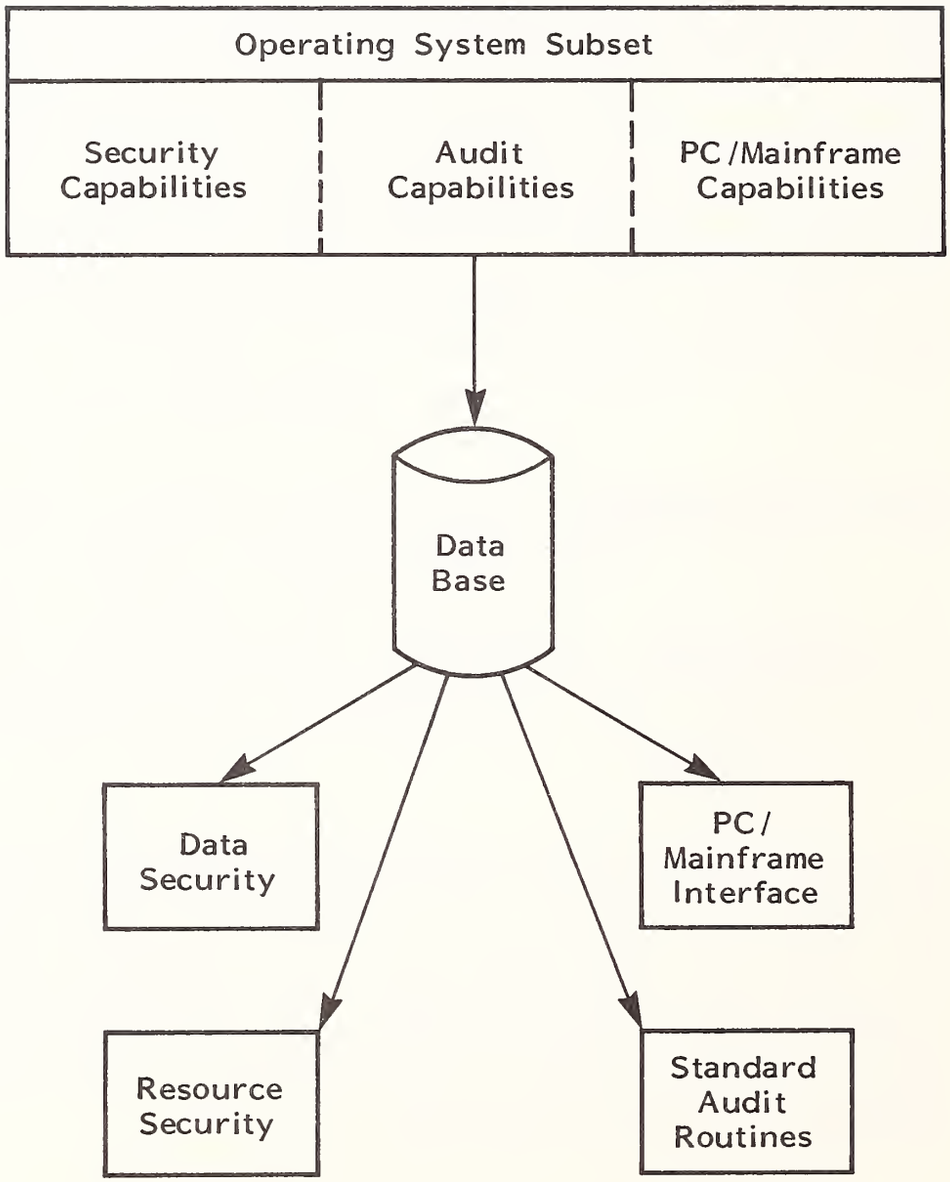
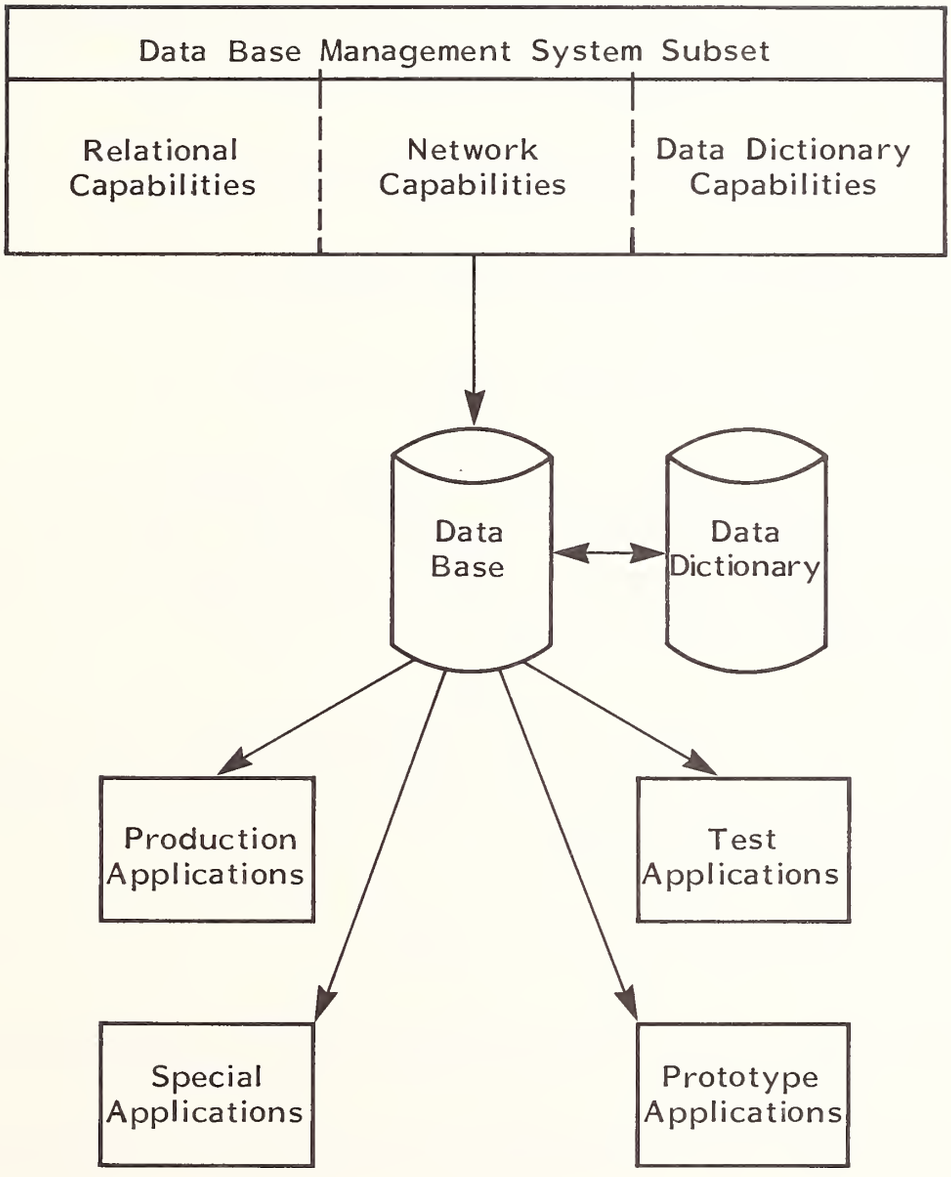


EXHIBIT V-3

DBMS SUPPORT TOOLS SUBSET



D. SYSTEMS DEVELOPMENT SUPPORT TOOLS

- A major effort is required to integrate existing and future systems development capability into today's 4GLs with special emphasis on automating the systems requirement phase of development. There are several automated tools now being marketed which represent various parochial approaches to systems definition, such as:

  - Excellerator, Index Technology Corporation.

  - Information Planner, Database Design, Inc.

  - Structures, Ken Orrand Associates, etc.

- Vendors may either opt for a strategic alliance with the most suitable products or build gateways to these products. Requirements definition is perhaps the least automated and the most cost critical function in the applications development process.

- Defining user requirements and translating them into a programmable specification requires an automated mechanism. The deliverables to meet these requirements should include:

  - Data inputs/outputs.

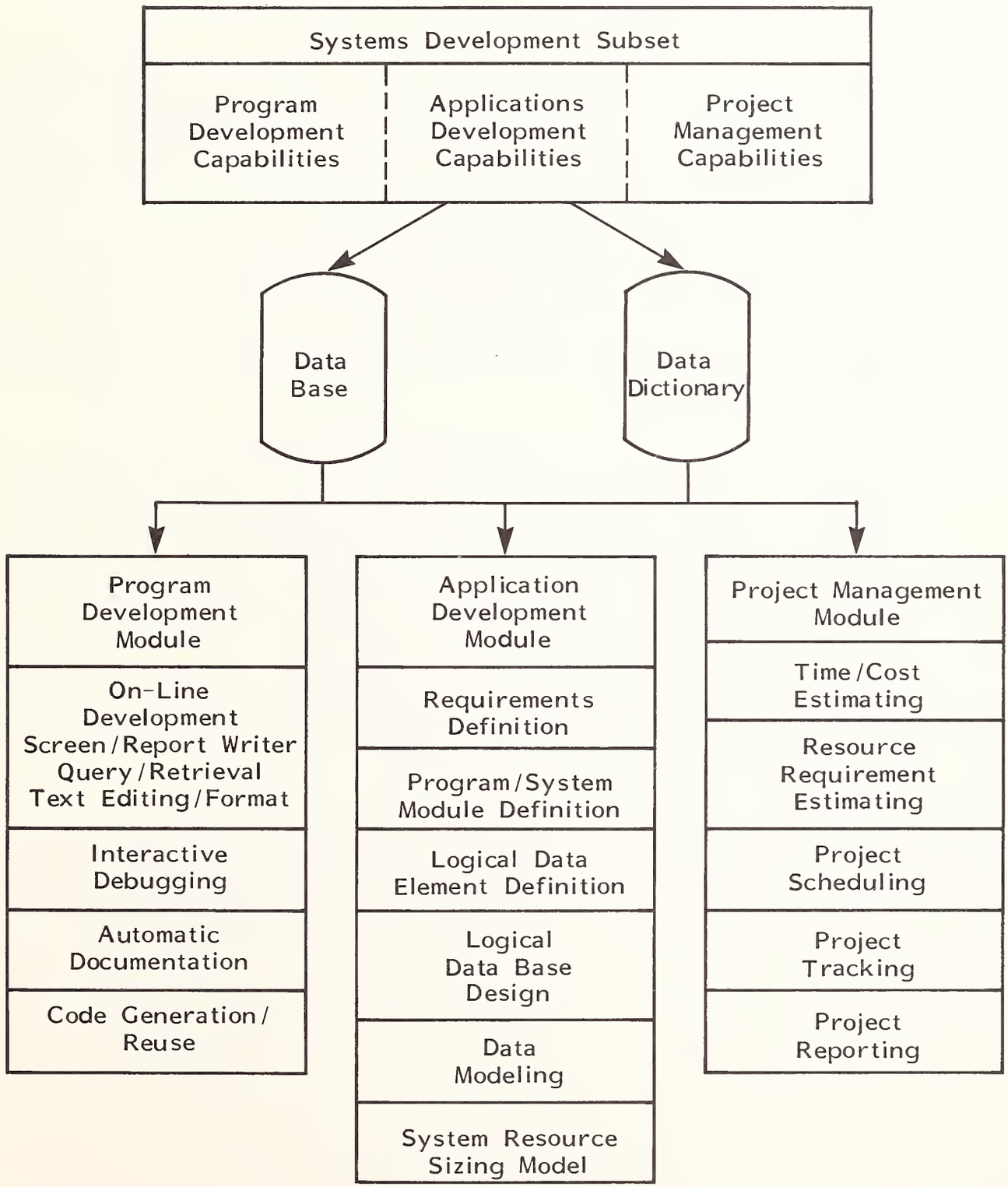
  - System, system modules, program modules, and program specification.

  - Logical data base definition to the record, segment, and field levels.

- Exhibit V-4 presents a view of the structure of these requirements. The three major modules required to support the emerging developer workstation requirements are program, applications, and project management capabilities.

EXHIBIT V-4

SYSTEMS DEVELOPMENT SUPPORT TOOLS SUBSET



- The program development module is well established in many of today's products. In fact, most of the productivity improvements cited in this report have been generated by these facilities. Integrating this module with the applications development and project management modules would close the loop and provide a full set of tools which are targeted to the most cost-intense activities in applications development.

- The applications development module would require a vehicle to translate user requirements from simulated business functions into preprogram definitions. This capability would allow for a definition of the application at the major system, subsystems, and program module levels. From this level of prototype detail, logical data elements and a data base specification can be developed. Data modeling and system resource sizing can then be prototyped and validated with the user. Detailed program specification and validation would follow this step, and prototyping would be improved to include a solid user requirements foundation.

- Project management tasks can be developed in the project management module to include time/cost and system (hardware/software) resource requirements estimates for project justification and tracking. Several respondents indicated that project on-time/on-cost performance still needs improvement. This link between user requirements, project resources, and project scheduling demand and project scheduling, tracking, and reporting would close the "applications development loop" and allow for improved performance.

- In summary, a 5GL that addressed the above requirements would satisfy the needs documented by the respondents and provide the basis for improved productivity throughout the entire applications development life cycle.

VI CONCLUSIONS AND RECOMMENDATIONS

A. CHOOSING A PROTOTYPING TOOL

- The selection of a prototyping tool should be based on a comprehensive plan for its use. This plan should include the following considerations:
  - Personnel training.
  - Integration with existing DBMS.
  - Pilot project selection.
  - Hardware, software, operating system compatibility.
- After developing a plan for use within the IS organization, a qualitative evaluation of vendor prototyping products should be performed. Exhibit VI-1 provides a working checklist based on the 23 features developed by this research. The 12 features which were found to be most important by the respondents are marked with an asterisk. Using this checklist, plus the definitions presented in Appendix A, will provide a basis for product comparisons.
- The final product selection should be made based on a thorough business analysis which isolates the product that:

EXHIBIT VI-1

VENDOR PROTOTYPING PRODUCTS CHECKLIST

	VENDOR PROTOTYPING PRODUCTS			
	A	B	C	ETC.
General Support Features				
Remote Job Entry				
Teleprocessing Monitor				
PC/Mainframe Link*				
Audit (File Footing, etc.)*				
Security (Data/Resource)*				
Optimized Compiler				
Linked Application Software				
High Quality Graphics				
OPS System Independent*				
Data Management Support Features				
Relational Data Base*				
Data Dictionary*				
Distributed Data Base Support				
Data/Language Independence*				
Program Development Features				
On-Line Development*				
Batch Development				
Interactive Debugging				
Code Generation/Reuse				
Automatic Documentation				
Query/Retrieval*				
Screen/Report Writer*				
English Language Query				
Foreign Language Conversion				
Text Editing/Formatting*				

*Represent the 12 most important prototyping features.

- Meets or exceeds the organization's planned use.

- Is favorably rated by the vendor's customer base.

- Is technically and organizationally compatible with the organization's environment.

- Choosing the right tool is a function of what it will be used for and then how best it meets the requirements. The sample of vendor products reviewed differs considerably in features offered. No attempt has been made to qualitatively judge products because user requirements and plans must be compared to tool capabilities.

B. CRITICAL PROTOTYPING SUCCESS FACTORS

- Four critical success factors have been isolated based on this research. These factors apply regardless of the complexity of the project and become more important with the spread of the technique's use. The critical success factors are:

  - The IS organization needs to plan thoroughly, allocate resources, and commit to the use of prototyping as an alternative to the traditional life cycle development process as a way of improving productivity and reducing the applications backlog.

  - It is crucial that users be an integral part of the prototyping process. Their responsibilities should include review and approval of key deliverables. They should be co-equal members of the development team.

- Prototyping benefits and degree of success should be measured by the reduction of project time required to deliver a user-approved operational system. It must be understood that it is a cooperative learning/doing process for both the user and developer.

- The selection and use of an efficient DBMS-oriented software product or suite of products is required to make prototyping practical.

C. PROTOTYPING STRENGTHS AND WEAKNESSES

- Prototyping clearly provides a significant opportunity to accrue improved applications development productivity. Improvements are reported to range from 62-283% over 3GLs through the use of existing 4GL products.

- This research has documented substantial improvements which are contributing to the respondents' ability to:

  - Chip away at the applications backlog.

  - Improve user/IS communications.

  - Implement successful applications to meet their firms' automation needs.

- Areas of specific improvements are listed in Exhibit VI-2. These strengths are presented in average rank order productivity gains that respondents reported.

- Client percentage improvements will obviously differ, but these values can be used as order of magnitude standards for those presently using or just initiating prototyping.

EXHIBIT VI-2

SPECIFIC PROTOTYPING BENEFITS REPORTED

PRODUCTIVITY GAINS (Rank Order)	BENEFITS	SAMPLE PERCENTAGE IMPROVEMENTS
1	Shorter time to build the first prototype.	283%
2	Reduction in the number of programming statements.	260
3	Program debugging time improvements related to performing programming mechanics such as compilations, opens, etc.	214
4	Better reliability and readability of application code.	148
5	Improvement in program modifications due to fewer syntax errors.	134
6	Training time it took to be productive.	123
7	Design improvement due to automatic code generation, e.g., data definitions, screen formatting, processing logic, etc.	96
8	Production systems improvement by using automated data editing features.	76
9	Increased ability to share program modules among applications.	62

- For those who have not yet purchased a prototyping tool, this research has isolated several deficiencies. In deciding on a vendor product, the following weaknesses should be considered:

  - Existing products require a high degree of technical IS knowledge for efficient use.

  - The next generation of prototyping products (referred to as 5GL) should be based on an expert system approach. Readers should look for an existing or new product which automates as many routine technical tasks as possible.

  - Extensive learning curves are required to use some of today's products. Less complicated, easier to use products should receive greater consideration.

  - A major capability that is lacking in today's products is the automation/integration of the applications requirements definition specification. The addition of this feature would provide a major productivity enhancement. This would clearly give a vendor a competitive edge and provide an additional degree of freedom for detailed design tasks.

- In summary, INPUT recommends that clients use this research as a benchmark for successful prototyping. For those who are not prototyping, it provides guidelines for success. For those who are involved in prototyping, the quantitative results and approaches can be used to compare results.

APPENDIX A: DEFINITIONS

A. GENERAL SUPPORT FEATURES

- AUDIT - Provides a set of standard routines for performing auditing tasks such as sampling, file footing, etc.

- HIGH QUALITY GRAPHICS - Provides the ability to extract data from a data base or system module and produces a variety of high resolution displays on a graphics terminal or printer in black and white or color.

- LINKED APPLICATION SOFTWARE - Provides users with business application packages that are compatible with a host data base.

- OPERATING SYSTEM INDEPENDENCE - Provides prototype applications which are insulated from operating system version changes.

- OPTIMIZED COMPILER - Provides a facility for automatically structuring code that minimizes the use of system resources.

- PC/MAINFRAME LINK - Provides a software link between a higher order processor and a personal computer to download data.

- REMOTE JOB ENTRY - Provides access to popular terminal devices through standard interfaces.

- SECURITY - Provides a working set of data and resource security measures to prevent unwanted access and alteration or removal of stored data.

- TELEPROCESSING MONITOR - Provides mainframe interface with major TP monitors for distributed processing.

B. DATA MANAGEMENT SUPPORT FEATURES

- DATA DICTIONARY - Provides a means to document the source and use of all data used within an application or system.

- DATA/LANGUAGE INDEPENDENCE - Provides a controlled correspondence between the logical structure and physical disposition of the data and the application (language).

- DISTRIBUTED DATA BASE SUPPORT - Provides an operating system feature that allows distribution of data and application models.

- RELATIONAL DATA BASE STRUCTURE - Provides for unstructured retrieval of a data element required for meeting ad hoc or unanticipated use of stored data. Offers some implementation of the evolving relational data base model.

C. PROGRAM DEVELOPMENT FEATURES

- AUTOMATIC DOCUMENTATION - Provides the ability to produce source listings, glossaries of definitions, etc., whenever the application design is updated, assuring that the documentation matches the current application version.

- BATCH DEVELOPMENT - Provides the facility of processing large data intense jobs such as validating transaction files or updating the data base.

- CODE GENERATION/REUSE - Provides the facility to develop program code and application module code that can be used as standard routines.

- ENGLISH LANGUAGE QUERY - Provides users with minimum data processing knowledge to access system, application, or data base information by using the English language.

- FOREIGN APPLICATION CONVERSION - Provides a software tool for converting non-vendor (foreign) applications to the DBMS used for prototyping.

- INTERACTIVE DEBUGGING - Provides program design capability of correcting code errors as the programs are being built.

- ON-LINE DEVELOPMENT - Provides real-time, interactive development, execution, and verification.

- QUERY/RETRIEVAL - Provides a conversational mode for use during prototyping to retrieve information from the data base or system files.

- SCREEN/REPORT WRITER - Provides the capability to develop, edit, and format CRT screens and hard copy reports.

- TEXT EDITING AND FORMATTING - Provides an integrated system facility for text editing and formatting during system program development.

APPENDIX B

USER QUESTIONNAIRE

APPLICATION SIMULATION AND PROTOTYPE TOOLS

Good morning (afternoon). This is _____ calling from INPUT, an international research and planning firm. We are currently engaged in a study of application development simulation and prototyping software tools. We would like a few minutes of your time to gain some basic information about your experience using this approach for software development. The information you provide will be held in complete confidence and will be used for statistical purposes only. Your identity and that of your firm will not be revealed. In appreciation for your help we will send you a summary of the report.

May we begin? Thank you.

1. First, we would like you to respond to the generic definitional elements of prototyping by rating the following attributes using a scale of one to five (1-5). On this scale "one" is unimportant while "five" is a very important part of your prototyping effort. How would you rate the importance of:

Interactive process for developing a final application specification

1 2 3 4 5 DK_____ NA_____ REF_____

2. Reduced time between iterations

1 2 3 4 5 DK_____ NA_____ REF_____

3. User involvement/approval for each version

1 2 3 4 5 DK_____ NA_____ REF_____

4. Initial prototype produced at minimum cost

1 2 3 4 5 DK_____ NA_____ REF_____

5. Integrated hardware/software support tools

1 2 3 4 5 DK_____ NA_____ REF_____

6. Are there any attributes which we have not covered?

YES_____ NO_____ DK_____ NA_____ REF_____

6a. IF YES: What might these be?

A.1 _____ 1 2 3 4 5 DK__ NA__ REF__

A.2 _____ 1 2 3 4 5 DK__ NA__ REF__

A.3 _____ 1 2 3 4 5 DK__ NA__ REF__

7. We would like to establish your experience level regarding application prototyping. For which of the following kinds of projects have you used prototyping:

Simple Reports YES___ NO___ DK___ NA___ REF___

8. Small Single Applications YES___ NO___ DK___ NA___ REF___

9. Large Single

Applications YES___ NO___ DK___ NA___ REF___

10. Small Integrated

Applications YES___ NO___ DK___ NA___ REF___

11. Large Integrated

Applications YES___ NO___ DK___ NA___ REF___

12. How long has prototyping been a formal part of your application development process?

_____ # Years

13. Now we would like to document your company's MIS budget, so we can measure the impact of your prototyping effort. First, what is the total MIS budget in millions?

1985 MIS budget _____ (Millions)

14. What is the per year expected growth rate for the next 3 years? _____ % per year

What percent of your 1985 budget is for:

15. Total hardware _____ %

16. Total purchased software _____ %

17. Total in-house developed software _____ %

18. Total communications services _____ %

19. Total other (100% of 14) _____ %

20. What percentage of both purchased and in-house software (questions 17 and 18) budget applies to prototyping?

_____ %

21. What percent growth rate <u>per year</u> would you estimate for all prototyping related activities for the next 3 years?

_____ %

22. Could you now list the prototyping tools you use, providing a vendor citation and rating for each? The rating is on a scale of 1-5, with 1 being inadequate for prototyping and 5 being adequate for prototyping.

	TOOLS	VENDOR							
T1.	_____	_____	1	2	3	4	5	DK__	NA__ REF__
T2.	_____	_____	1	2	3	4	5	DK__	NA__ REF__
T3.	_____	_____	1	2	3	4	5	DK___	NA__ REF__
T4.	_____	_____	1	2	3	4	5	DK__	NA__ REF__
T5.	_____	_____	1	2	3	4	5	DK___	NA__ REF__

23. Are there any comments you would like to make which would explain your ratings?

YES_____ NO_____ DK_____ NA_____ REF_____

24 If Yes, what might these be? (INT. note tool from Q23).

_____

_____

_____

25. Now we would like to discuss the features included in your support tools and the technical knowledge required to use these features. Please rate the importance of the following features using a 1-5 scale where 1 is unimportant and 5 is a very important prototyping feature tool. Please also rate the amount of technical IS knowledge required, with 1 being no technical IS knowledge and 5 being advanced technical IS knowledge.

The first set of items refers to General Support features. How important would you rate:

		FEATURE	KNOWLEDGE
26.	Remote Job Entry?	1 2 3 4 5 DK NA REF	1 2 3 4 5 DK NA REF
27.	Teleprocessing Monitor?	1 2 3 4 5 DK NA REF	1 2 3 4 5 DK NA REF
28.	PC Tool Linked to Mainframe?	1 2 3 4 5 DK NA REF	1 2 3 4 5 DK NA REF
29.	Audit, e.g., confirmation, file footing, sampling, etc.?	1 2 3 4 5 DK NA REF	1 2 3 4 5 DK NA REF
30.	Security, e.g., data and resource?	1 2 3 4 5 DK NA REF	1 2 3 4 5 DK NA REF

		FEATURE	KNOWLEDGE
31.	Optimized Compiler?	1 2 3 4 5 DK NA REF	1 2 3 4 5 DK NA REF
32.	Linked Application Software?	1 2 3 4 5 DK NA REF	1 2 3 4 5 DK NA REF
33.	High-Quality Graphics?	1 2 3 4 5 DK NA REF	1 2 3 4 5 DK NA REF
34.	Operating System Independence?	1 2 3 4 5 DK NA REF	1 2 3 4 5 DK NA REF

Turning to the area of Data Management Support facilities how would you rate:

35.	Relational DBMS?	1 2 3 4 5 DK NA REF	1 2 3 4 5 DK NA REF
36.	Data Dictionary?	1 2 3 4 5 DK NA REF	1 2 3 4 5 DK NA REF

		FEATURE	KNOWLEDGE
37.	Distributed DB Support?	1 2 3 4 5 DK NA REF	1 2 3 4 5 DK NA REF
38.	Data and Language Independence?	1 2 3 4 5 DK NA REF	1 2 3 4 5 DK NA REF

The next set of items refer to Program Development Support features. How would you rate:

39.	On-line/Batch Development?	1 2 3 4 5 DK NA REF	1 2 3 4 5 DK NA REF
40.	Interactive Debugging?	1 2 3 4 5 DK NA REF	1 2 3 4 5 DK NA REF
41.	Code Generation and Reuse?	1 2 3 4 5 DK NA REF	1 2 3 4 5 DK NA REF

		FEATURE	KNOWLEDGE
42.	Automatic Documentation?	1 2 3 4 5 DK NA REF	1 2 3 4 5 DK NA REF
43.	Query/Retrieval?	1 2 3 4 5 DK NA REF	1 2 3 4 5 DK NA REF
44.	Screen and Report Writer?	1 2 3 4 5 DK NA REF	1 2 3 4 5 DK NA REF
45.	English Language Query?	1 2 3 4 5 DK NA REF	1 2 3 4 5 DK NA REF
46.	Foreign Application Conversion?	1 2 3 4 5 DK NA REF	1 2 3 4 5 DK NA REF
47.	Text Editing and Formatting?	1 2 3 4 5 DK NA REF	1 2 3 4 5 DK NA REF

48. We would like to measure the benefits accruing from your current application prototyping activities. Comparing your current prototyping efforts to your third generation language (3GL) experience, what percentage improvement are you experiencing in the following areas:

Number of training hours to be productive

% _____ DK___ NA___ REF___

49. Building the 1st prototype

% _____ DK___ NA___ REF___

50. Length of Design Phase

% _____ DK___ NA___ REF___

51. Length of Development Phase

% _____ DK___ NA___ REF___

52. Length of Implementation Phase

% _____ DK___ NA___ REF___

53. Entire life cycle effort (Design to Implementation) is shorter

% _____ DK___ NA___ REF___

54. Ability to share program modules among applications

% _____ DK___ NA___ REF___

55. Time used for Automatic Application Code Generation, for Data Definitions, Screen Formating and Processing logic,

% _____ DK___ NA___ REF___

56. Overall reduction of the Number of Programming Statements

% _____ DK___ NA___ REF___

57. Reduction of Program modifications due to syntax errors

% _____ DK___ NA___ REF___

58. Reduction of programming mechanics, such as compilations, opens, etc. that results in faster debugging

% _____ DK___ NA___ REF___

59. Readability and reliability of applications code

% _____ DK___ NA___ REF___

60. Automated Data Editing

% _____ DK___ NA___ REF___

61. Ease of Maintenance Changes/Updates

% _____ DK___ NA___ REF___

62. Given your prototyping experience and the benefits you have described, what are your thoughts regarding the weaknesses of the current prototyping technology? (INT. Please list and rate the weaknesses on a scale of one to five with one being minor and five being major.)

	WEAKNESSES	APPROACH
W1.	_____	1 2 3 4 5
W2.	_____	1 2 3 4 5
W3.	_____	1 2 3 4 5
W4.	_____	1 2 3 4 5
W5.	_____	1 2 3 4 5

63. Given your current prototyping experience, will your approach change in the future?

Y =1 N =2 DK_____ NA_____ REF_____

63a. Why is this? _____

_____

_____

64. Will there be increased use in more complicated projects?

Y =1 N =2 DK_____ NA_____ REF_____

64a. Why is this? _____

_____

_____

_____

_____

65. Would new application areas be considered?

Y =1 N =2 DK_____ NA_____ REF_____

65a. Why is this? _____

_____

_____

_____

_____

66. Are there any other plans you have for prototyping that we have not covered?

YES_____ NO_____ DK_____ NA_____ REF_____

66a. If yes, what are they?

_____

_____

_____

67. Reflecting on our conversation, are there any major IS or vendor issues, problems, concerns, that you would like to mention?

YES_____ NO_____ DK_____ NA_____ REF_____

67a. If Yes, what are they? _____

_____

_____

_____

Thank you.

APPENDIX C

SELECTED PROTOTYPING VENDOR PRODUCTS

COMPANY	PRODUCT NAME	TOTAL SALES ($ Millions)	NUMBER OF EMPLOYEES	HARDWARE	OPERATING SYSTEM	LANGUAGES	PRODUCT FIRST SOLD	AVERAGE PURCHASE PRICE ($ Thousands)	NUMBER OF PRODUCTS SOLD
Analysts International Corporation	Corvet	$44	750	Data General/VS	AOS/VS	–	1984	$45–$100	BETA TEST
Information Resources Incorporated	Express E300	75	900	IBM/Prime-4300, 50 Series	VM, CMS, PRIMOS	Answer DB INTER	1977	$65–$120	120
International Business Machines System 38	QIDU/DFU	40,200	369,000	System 38	CPF	RPG III, COBOL	1980	–	–
Execucom Systems Corporation	IFPS	25	200	Burroughs, CDC, Data General, DEC, Honeywell Information Systems, Hewlett-Packard, IBM, Prime, etc.	VM, CMS, etc.	Fortran, Assembly	1972	$40–$100	1,400 +
Cincom Systems Incorporated	MANTIS	100	1,450	IBM 370 Up, DEC VAX, Wang, Honeywell Information Systems, Bull, ICL	VM, MVS, VSE	IBM Callable	1981	$30–$ 85	2,000
Thorn-EMI Computer Software Incorporated	FCS	40	570	IBM 4300 Up, DEC VAX, Data General, MV, Hewlett-Packard 3000, Wang VS	CMS, TSO, CICS	FCS	1973	$50–$150	1,000 +
Information Builders Incorporated	FOCUS	72	425	IBM 4300 Up, DEC VAX, PC-Compatibles	MVS, TSO, CMS	COBOL, PL1, Assembly	1976	$66–$110	1,750
Informatics General Corporation	Mark V	200	2,000	IBM 4300 Up,	MVS, XA, VSE, VM, CMS, CICS, IMS DL	IBM Callable	1981	$30–$100	250
Martin Marietta Data Systems	UFO	–	90	IBM 370 Up	IBM Compatible	COBOL, Assembly	1976	$28–$ 40	1,700
Software Express Incorporated	APPGEN	–	60	UNIX, XENIX	UNIX, XENIX	"C"	1982	–	–
Comshare Incorporated	System W	62	800	IBM 4300 Up	VM, CMS, MVS, TSO	Pascal	1982	$55–$198	400 +
Cullinet Software Incorporated	ADS/ONLINE	79	814	IBM 360/70, Siemens 75/77	MVT, MFT, VS1	–	–	–	–
Software AG Incorporated	Natural	120	1,00	IBM 4300 Up	VM, MVS, XA, VS1, DOS, VSE	IBM Standard Calls	1979	$30–$ 60	1,500

About INPUT

INPUT provides planning information, analysis, and recommendations to managers and executives in the information processing industries. Through market research, technology forecasting, and competitive analysis, INPUT supports client management in making informed decisions. Continuing services are provided to users and vendors of computers, communications, and office products and services.

The company carries out continuous and in-depth research. Working closely with clients on important issues, INPUT's staff members analyze and interpret the research data, then develop recommendations and innovative ideas to meet clients' needs. Clients receive reports, presentations, access to data on which analyses are based, and continuous consulting.

Many of INPUT's professional staff members have nearly 20 years' experience in their areas of specialization. Most have held senior management positions in operations, marketing, or planning. This expertise enables INPUT to supply practical solutions to complex business problems.

Formed in 1974, INPUT has become a leading international planning services firm. Clients include over 100 of the world's largest and most technically advanced companies.

Offices

NORTH AMERICA

Headquarters
1943 Landings Drive
Mountain View, CA 94043
(415) 960-3990
Telex 171407

New York
Parsippany Place Corp. Center
Suite 201
959 Route 46 East
Parsippany, NJ 07054
(201) 299-6999
Telex 134630

Washington, D.C.
11820 Parklawn Drive
Suite 201
Rockville, MD 20852
(301) 231-7350

EUROPE

United Kingdom
INPUT
41 Dover Street
London W1X 3RB
England
01-493-9335
Telex 27113

Italy
Nomos Sistema SRL
20127 Milano
Via Soperga 36
Italy
Milan 284-2850
Telex 321137

Sweden
Athena Konsult AB
Box 22232
S-104 22 Stockholm
Sweden
08-542025
Telex 17041

ASIA

Japan
ODS Corporation
Dai-ni Kuyo Bldg.
5-10-2, Minami-Aoyama
Minato-ku,
Tokyo 107
Japan
(03) 400-7090
Telex 26487

INPUT®
Planning Services For Management

INPUT®
1943 Landings Drive, Mountain View, CA 94043, (415) 960-3990