THE OPPORTUNITIES OF FOURTH GENERATION LANGUAGES

INPUT

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, commur
and services.

The company carries
research. Working cl
tant issues, INPUT's
interpret the researc
mendations and inr

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
in operations, marketing, or planning.
rtise enables INPUT to supply practical
o complex business problems.

1974, INPUT has become a leading
planning services firm. Clients include
he world's largest and most techni-
mpanies.

OFFICES

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

Detroit
220 E. Huron
Suite 209
Ann Arbor, MI 4810
(313) 971-0667

New York
Park 80 Plaza West-1
Saddle Brook, NJ 076
(201) 368-9471
Telex 134630

United Kingdom
INPUT, Ltd.
Airwork House
35 Piccadilly
London, W1V 9PB
England
01-439-8985
Telex 23116

n
seas Data Service Company, Ltd.
etsu Building
2 - 7 Kita Aoyama
ome Minato-ku
yo, 107
n
400-7090
x 26487

den
ena Konsult
Persson & Co AB
22114
4 22 Stockholm
den
08-52 07 20
Telex 17041

Telex 220064 X5533

Italy
PGP Sistema SRL
20127 Milano
Via Soperga 36
Italy
Milan 284-2850
Telex 310352

West Germany
NOVOTRON GmbH
Am Elizabethenbrunnen 1
D-6380 Bad Homburg
West Germany
Telex 418094

INPUT Planning Services for Management

THE OPPORTUNITIES OF FOURTH GENERATION LANGUAGES

SEPTEMBER 1983

THE OPPORTUNITIES OF FOURTH GENERATION LANGUAGES

CONTENTS

		Page
I	INTRODUCTION	1
	A. Reasons For Preparing This Report	1
	B. Scope And Methodology	2
	C. Other Related INPUT Reports	3
II	EXECUTIVE SUMMARY	5
	A. The Opportunities Of Fourth Generation Languages	6
	B. How Third And Fourth Generation Languages Differ	8
	C. The Growth Of FGL Use Will Be High	10
	D. Fourth Generation Languages Reduce Both Visible And Invisible Backlogs	12
	E. FGLs Are The Emerging Link Between Micro And Mainframe Computers	14
	F. FGLs Correct Flaws In Traditional Systems Development	16
	G. The Costs Of Using FGLs In A Production Environment Can Vary Greatly	18
	H. FGLs Have More Positive Than Negative Factors Impelling Their Use	20
	I. FGLs May Have A Major Impact On Information Systems Staff	22
	J. Fourth Generation Language Provides Routes To Achieving Information Systems Goals	24
III	WHAT IS A FOURTH GENERATION LANGUAGE?	27
	A. General Description	27
	B. Selected Fourth Generation Language Characteristics	30
	1. FOCUS	30
	2. NOMAD2	32
	3. RAMIS II	33
	4. INQUIRE	35
	5. ADABAS/NATURAL	36
	6. ADS/OnLine Plus OnLine English Plus Information Database	38
	7. Mark V	40
	8. System W	41
	9. SALVO	42

		Page
IV	THE FOURTH GENERATION LANGUAGE ENVIRONMENT	45
	A. FGL Use	45
	1. Secondary FGL Uses	45
	2. Mainline Uses	47
	3. FGL Growth Expectations	48
	B. Fourth Generation Language Economics	52
V	THE USE OF FOURTH GENERATION LANGUAGES BY INFORMATION SYSTEMS DEPARTMENTS	63
	A. Prototyping	66
	B. FGLs In Production	72
	C. Mainframe Personal Computer FGL Linkage	76
	D. FGLs And System Quality	80
VI	IMPACTS OF FOURTH GENERATION LANGUAGES	85
	A. FGL Benefits	88
	B. FGL Impact On Personnel	92
	C. FGL Organizational Impact	96
VII	RECOMMENDATIONS FOR A FOURTH GENERATION LANGUAGE STRATEGY	101
	A. Strategic Issues	101
	B. Tactical Issues	104
APPENDIX A:	QUESTIONNAIRE	107
APPENDIX B:	ANALYTIC DESCRIPTIONS OF SELECTED FOURTH GENERATION LANGUAGE PACKAGES	117
APPENDIX C:	QUANTIFYING THE COST EFFECTS OF FOURTH GENERATION LANGUAGE USE IN A PRODUCTION ENVIRONMENT.....	157

THE OPPORTUNITIES OF FOURTH GENERATION LANGUAGES

EXHIBITS

			Page
II	-1	The Opportunities Of Fourth Generation Languages	7
	-2	How Third And Fourth Generation Languages Differ	9
	-3	The Growth Of FGL Use Will Be High	11
	-4	Fourth Generation Languages Reduce Both Visible And Invisible Backlogs	13
	-5	FGLs Are The Emerging Link Between Micro And Mainframe Computers	15
	-6	FGLs Correct Flaws In Traditional Systems Development	17
	-7	The Costs Of Using FGLs In A Production Environment Can Vary Greatly	19
	-8	FGLs Have More Positive Than Negative Factors Impelling Their Use	21
	-9	FGLs May Have A Major Impact On Information Systems Staff	23
	-10	Fourth Generation Languages Provide Routes To Achieving Information Systems Goals	25
III	-1	Types Of Fourth Generation Language Products	29
IV	-1	Classification Of Fourth Generation Language Use	49
	-2	Number Of Installations Of Selected Fourth Generation Language Products	50
	-3	Expected Increases In Fourth Generation Language Use, 1983-1985	51
	-4	Fourth Generation Language Cost Factors	54
	-5	Potential Cost Effects Of Fourth Generation Language Use In A Production Environment	56
	-6	Cost Impact Of Fourth Generation Language Use At Different Staffing Levels	57
	-7	Installed MIPS After Conversion To Fourth Generation Language	58
	-8	Installed MIPS Per Development Person After Fourth Generation Language Installation	60
V	-1	IS Views On The Use Of Fourth Generation Languages	64
	-2	A Fourth Generation Language Manufacturing Application	65
	-3	The Vicious Circle Of Traditional Software Development	67

			Page
	-4	Analysts - Weak Links And Bottlenecks?	69
	-5	Prototyping Effects On System Activities	71
	-6	Fourth Generation Language Conversion Targets	74
	-7	Information Center And Personal Computer Isolation	77
	-8	The Productivity Pyramid	81
VI	-1	FGL User Satisfaction With Information Systems	86
	-2	Information Systems' Fears Of FGLs	87
	-3	Fourth Generation Language Capabilities' Impact On Potential Benefits	91
	-4	Current Personnel Advancement	94
	-5	Post-FGL Personnel Relationships	95
	-6	Perspective Of Information Systems/User Personnel Movement: Information Systems Management	97
	-7	Perspective Of Information Systems/User Personnel Movement: Key Users	98
VII	-1	Positive, Ascending Feedback With FGL	103
	-2	Phased FGL Implementation For IS Projects	105
B	-1	Analytic Description Of Focus	118
	-2	Analytic Description Of Nomad 2	122
	-3	Analytic Description Of Ramis II	126
	-4	Analytic Description Of Adabas/Natural	130
	-5	Analytic Description Of Inquire	134
	-6	Analytic Description Of Ads And Ole	138
	-7	Analytic Description Of Mark V	142
	-8	Analytic Description Of System W	146
	-9	Analytic Description Of Salvo	150
	-10	Vendor Names And Addresses	154
C	-1	Fourth Generation Cost Impact On IS Expense Patterns	159
	-2	Fourth Generation Cost Impact On IS Expense Patterns	161
	-3	Fourth Generation Cost Impact On IS Expense Patterns	162

1 INTRODUCTION

I INTRODUCTION

A. REASONS FOR PREPARING THIS REPORT

- Fourth generation languages (FGLs) may be nearing the takeoff stage where they will have significant impact on many companies, in both information systems (IS) and non-information systems areas.

  - FGLs, if introduced and controlled correctly, can increase productivity in data processing as well as in underlying business activities.

  - On the other hand, if FGL use is not planned, it may have only marginal benefits or even add overhead costs and other inefficiencies to an organization.

- FGLs are now in a period of transition, which makes this report particularly opportune.

  - New features and new capabilities are being added to established, usually mainframe-based FGLs.

  - New FGLs are being introduced, many aimed at the personal computer (PC) user.

- Many of the established products have introduced or are planning to introduce mainframe/PC-linked products. These could have significant impact on corporations and on IS departments in particular.

- Most important, FGLs may be on the brink of moving from primarily an end-user-oriented tool (e.g., via the Information Center) to one that will be a major means for accomplishing mainline IS tasks.

B. SCOPE AND METHODOLOGY

- This report is part of the Information Systems Program (ISP) and addresses the following issues:

  - Definition and description of FGLs (Chapter III).

  - The FGL environment, especially FGL economics (Chapter IV).

  - The applicability of FGLs to IS itself (Chapter V).

  - The impact of the FGL (Chapter VI).

  - The recommended IS strategy (Chapter VII).

- The information for this report was gathered from the following sources:

  - Thirty-eight structured interviews with personnel from a variety of companies using or planning to use an Information Center (IC). The survey instrument is in Appendix A.

  - Over a dozen in-depth interviews with leading FGL vendors.

- INPUT's studies about software productivity.

- INPUT also has discussed these issues in depth with over a dozen leading IS practitioners.

- INPUT has taken the best practices and proposals and subjected them to further analysis to serve as the basis for the analysis and recommendations in this report.

C. OTHER RELATED INPUT REPORTS

- Readers of this report are advised to review two earlier INPUT reports:

  - Personal Computers in the IS Strategy, December 1982.

    . This report recommends the most effective ways for IS to become involved with PCs.

  - Supporting Personal Computer Software, August 1983.

    . As FGLs become PC-based, the issues of ongoing PC software support become important in the successful use of FGLs.

  - Organizing the Information Center, August 1983.

    . Currently (and for the foreseeable future) the primary FGL delivery vehicle is the Information Center. Consequently, the FGL and Information Center strategies and support should be well coordinated.

II EXECUTIVE SUMMARY

II EXECUTIVE SUMMARY

- Note: this executive summary is designed in a presentation format in order to:

  - Help the busy reader quickly review key research findings.

  - Provide a ready-to-go executive presentation, complete with a script, to facilitate group communication.

- The key points of the entire report are summarized in Exhibits II-1 through II-10. On the left-hand page facing each exhibit is a script explaining its contents.

A. THE OPPORTUNITIES OF FOURTH GENERATION LANGUAGES

- This research was produced as a part of INPUT's Information Systems Program (ISP).

- INPUT believes that fourth generation languages (FGLs) will have a significant impact on data processing and end-user productivity. But if FGL use is not properly planned and controlled, it may produce only marginal benefits, while increasing costs and inefficiencies throughout the organization.

- INPUT's research report:

  - Defines and describes fourth generation languages.

  - Describes the applicability of using FGLs for information systems (IS) development.

  - Discusses the impact fourth generation languages have on the organization.

  - Provides recommended IS strategies to fully utilize the capabilities of FGLs.

- The remainder of this presentation will provide highlights from INPUT's report.

EXHIBIT II-1

THE OPPORTUNITIES OF FOURTH GENERATION LANGUAGES

- INPUT ISP report
- Impact of FGLs
  - Improve data processing and end-user productivity
  - Lack of control may produce negative results
    - Marginal benefits
    - Increased costs and inefficiencies
- Research scope
  - Defines FGL
  - IS uses
  - Impact on organization
  - Recommendations

B. HOW THIRD AND FOURTH GENERATION LANGUAGES DIFFER

- Fourth generation languages (FGL) differ from third generation languages (e.g., COBOL) in that they:

  - Are oriented to nonprogrammers as well as programmers and consequently use an English-oriented, nonprocedural approach.

  - Can be learned quickly (i.e., in a few hours or, perhaps, a day or two) to use for simpler applications.

  - Most important, they can accomplish a given task five or ten times faster than third generation languages.

- FGLs have been most accepted and used in end-user environments, often in an Information Center setting. End users can:

  - Produce reports from already existing files or data bases.

  - Write their own programs or production systems.

- More ambitiously, FGLs can also be used in conjunction with "main-line" system development, either by:

  - Prototyping (and then writing third generation codes).

  - Executing FGL code in a production setting.

EXHIBIT II-2

HOW THIRD AND FOURTH GENERATION LANGUAGES DIFFER

CHARACTERISTICS	THIRD GENERATION LANGUAGES	FOURTH GENERATION LANGUAGES
Approach used	Procedural (i.e., How To Do It)	Nonprocedural (i.e., What Do You Want?)
Users	Programmers	Programmers or Nonprogrammers
Built-in Functions	Few	Many (e.g., Statistical, Graphics, Financial Functions)
Internal Data Base Function	No	Often
Quick Start-up	No	Yes
Used for Large, Complex Tasks	Yes	Sometimes

C. THE GROWTH OF FGL USE WILL BE HIGH

- Respondent FGL users and prospective users believe that the amount of use by programmers and nonprogrammers alike will increase at a very high rate (although sometimes from a low base) in the next two years.

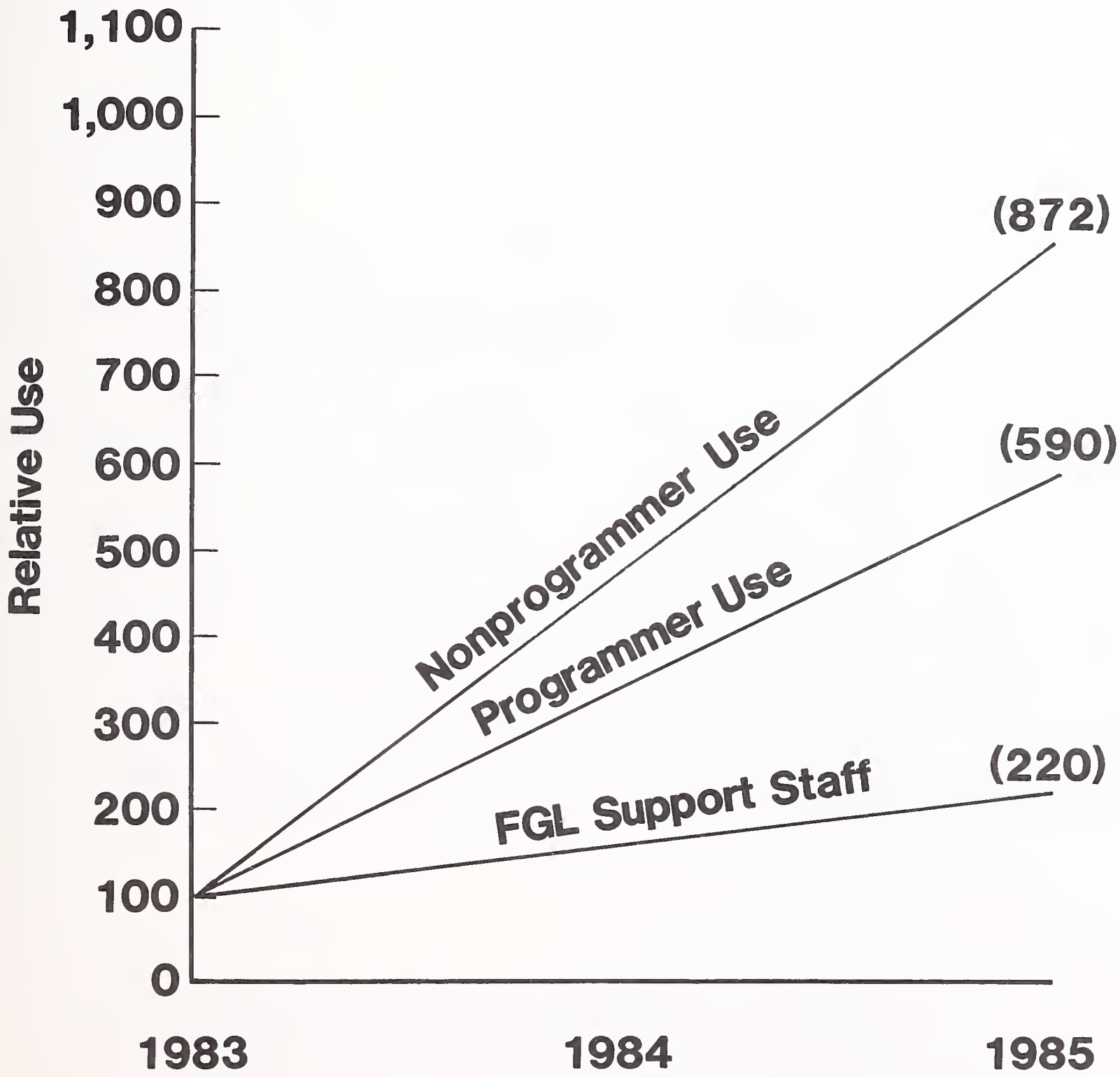
  - End-user use is expected to increase rapidly, reflecting FGL's historic use in the user community.

  - The high rate of use foreseen by IS programmers may signify increasing acceptance of FGLs for "professional" programming.

- A disquieting note is that FGL support is projected to lag considerably behind FGL use by both programmers and nonprogrammers. This same tendency was observed in INPUT's earlier study on Information Centers. Eventually, this lack of support could undermine the effective use of FGLs.

EXHIBIT II-3

THE GROWTH OF FGL USE WILL BE HIGH



1983 use = 100

D. FOURTH GENERATION LANGUAGES REDUCE BOTH VISIBLE AND INVISIBLE BACKLOGS

- The "visible" backlog of applications development is three years or more in many companies. This visible backlog usually consists of formal requests for large-scale development efforts.

- There is also an "invisible" backlog of needed applications, those that users see no point in submitting for approval once the visible backlog has grown too large.

  - The invisible backlog is probably the same size as the visible backlog.

  - However, it is generally made up of a significantly larger number of much smaller applications. Consequently, the hidden backlog is very attractive for end users to attack themselves via FGLs.

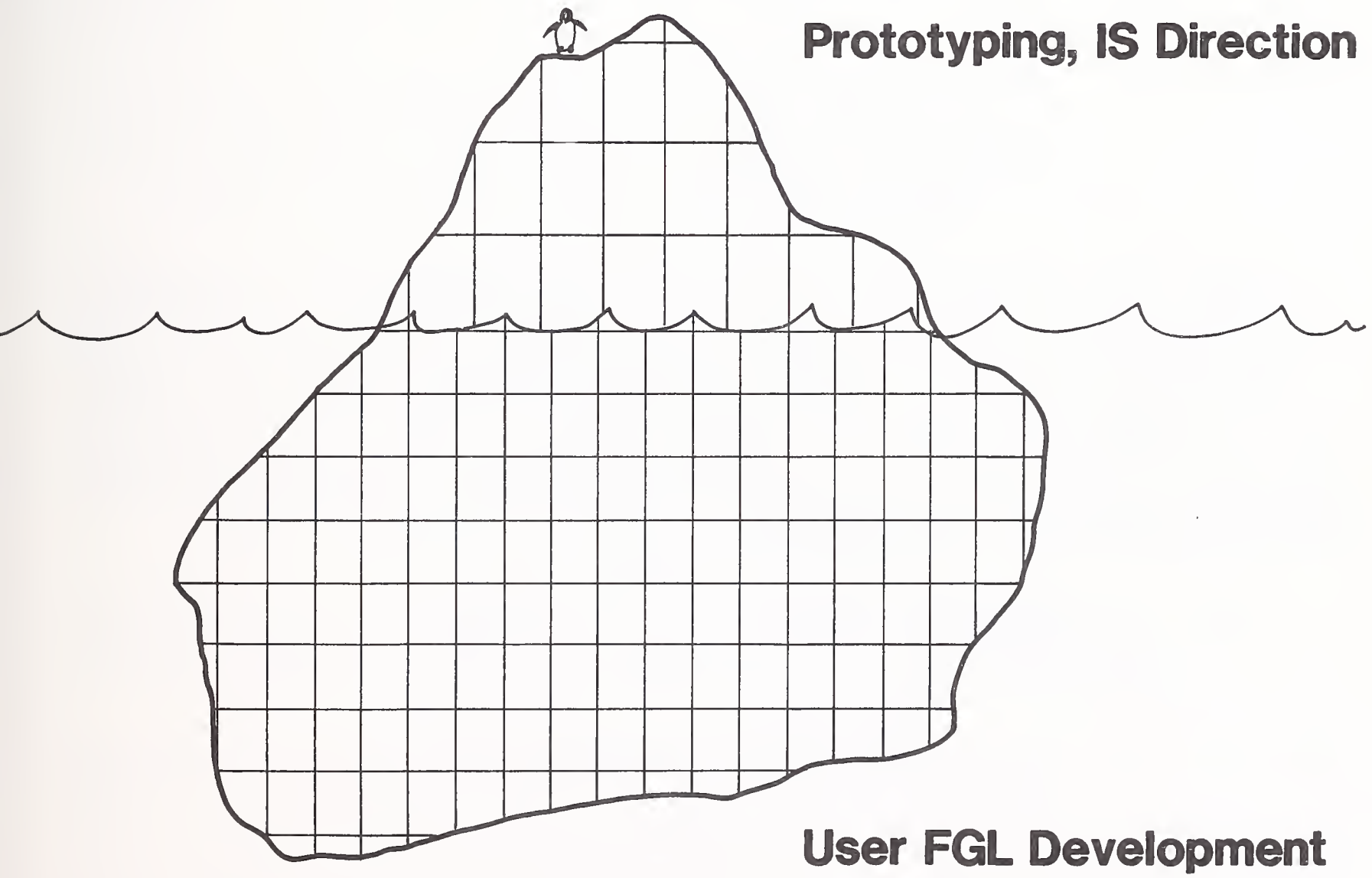
- The visible backlog of larger, more complex systems can be jointly attacked by IS and users using FGL.

  - Such joint FGL development should be undertaken where there is a definite system "owner" who is willing to assume development and ongoing maintenance responsibility.

  - Promising systems for FGL targeting are those with a high degree of changing user needs. An FGL can deal with this kind of change much better than conventionally developed systems can.

EXHIBIT II-4

FOURTH GENERATION LANGUAGES REDUCE BOTH VISIBLE AND INVISIBLE BACKLOGS



Relative Application Size

E. FGLs ARE THE EMERGING LINK BETWEEN MICRO AND MAINFRAME COMPUTERS

- FGLs have until very recently been mostly associated with large mainframes (packages like VisiCalc are partial exceptions).

  - However, in the past six months there have been several product announcements of downsized FGLs that will run on a personal computer.

  - Few FGL vendors will now say that they do not have a micro-FGL in development. It will soon be a matter of market survival for a current FGL vendor to have an FGL that will run on a personal computer.

- It is too early to say with certainty how transparent the new environment will be to a newly expanded FGL. Certainly all FGL vendors intend for there to be both upward and downward compatibility as well as transferability between the mainframe and the personal computer for a given FGL. But there will probably be a price to pay for this compatibility: inefficiency in operation and/or increased complexity (hence, difficulty in learning and use).

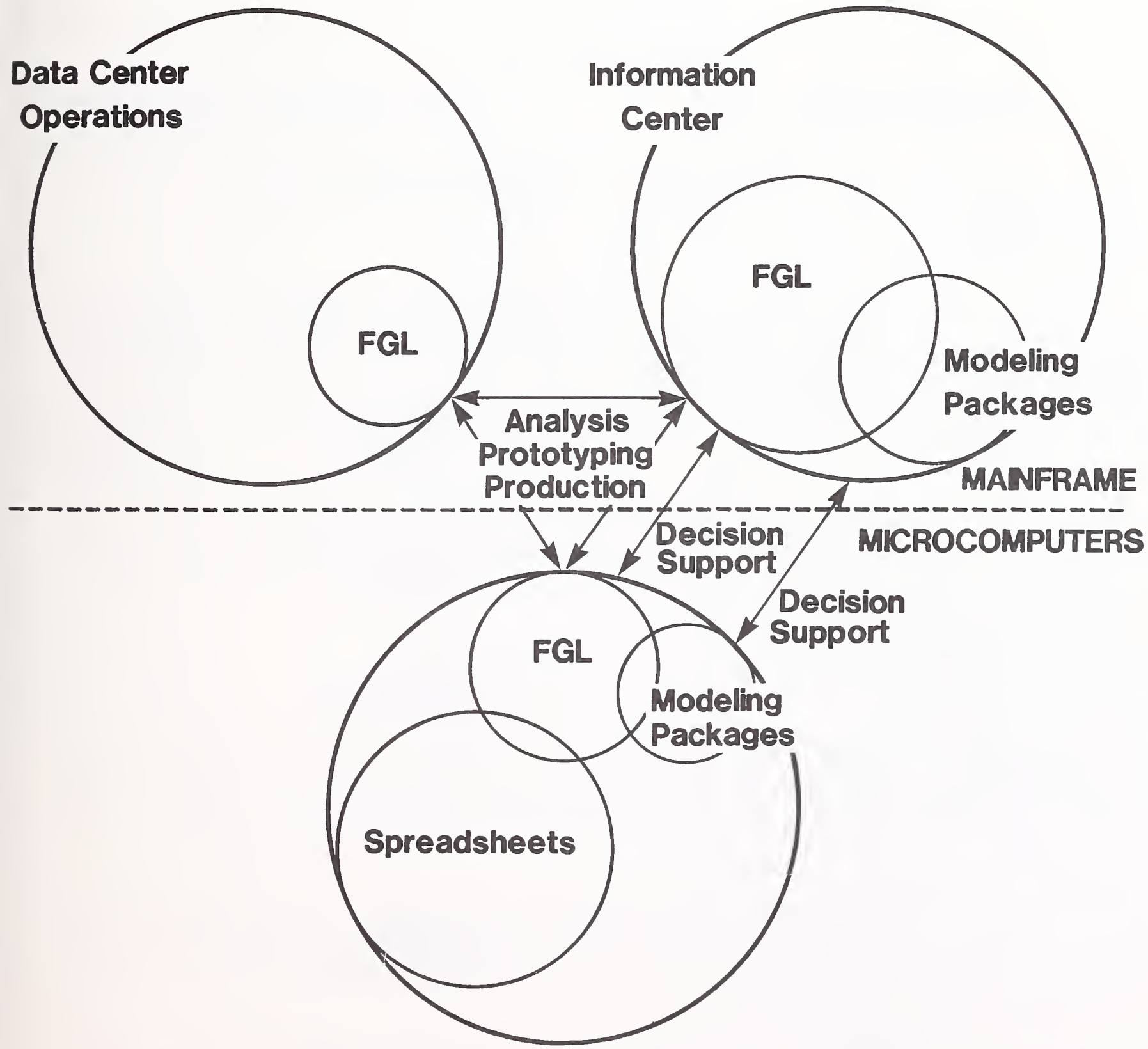
- However, assuming that the tradeoffs are reasonable, micro-FGLs offer the IS department many attractive options that were not fully available in the past.

  - Mainframes and personal computers can be linked in an applications sense, not as they are now, by means of file exchange. This will ease operation crunches by allowing applications to be developed and/or run on whatever machine media makes sense at the time (always keeping in mind current personal computer storage constraints).

  - A real payoff will come in the system-building mode where FGL applications can be prototyped and developed in a personal computer and/or information center environment and then moved, unchanged, into a production environment.

EXHIBIT II-5

FGLs ARE THE EMERGING LINK BETWEEN MICRO AND MAINFRAME COMPUTERS (1984-1985)



Links are shown as ⟶

F. FGLs CORRECT FLAWS IN TRADITIONAL SYSTEMS DEVELOPMENT

- Most respondents to INPUT's survey do not view FGL as the foundation for large, transaction-oriented systems: Most IS professionals see smaller, decision support-type systems as the natural strength of FGLs.

- Actually, the traditional life-cycle approach to systems development has several serious flaws, which can in large measure be overcome using an FGL.

  - The traditional approach is essentially a straight-line process. There is little or no iteration. It is assumed that tasks will be done correctly the first time.

  - The analysis phase tends to be lengthy and extremely detailed. Its length and detail makes it very difficult for end users to participate meaningfully. Too often, they must sign off on a large quantity of imperfectly understood written specifications.

  - The very time involved works against traditional development: by the time it is finished, users' requirements have often changed significantly.

- FGL-based development has three basic strengths:

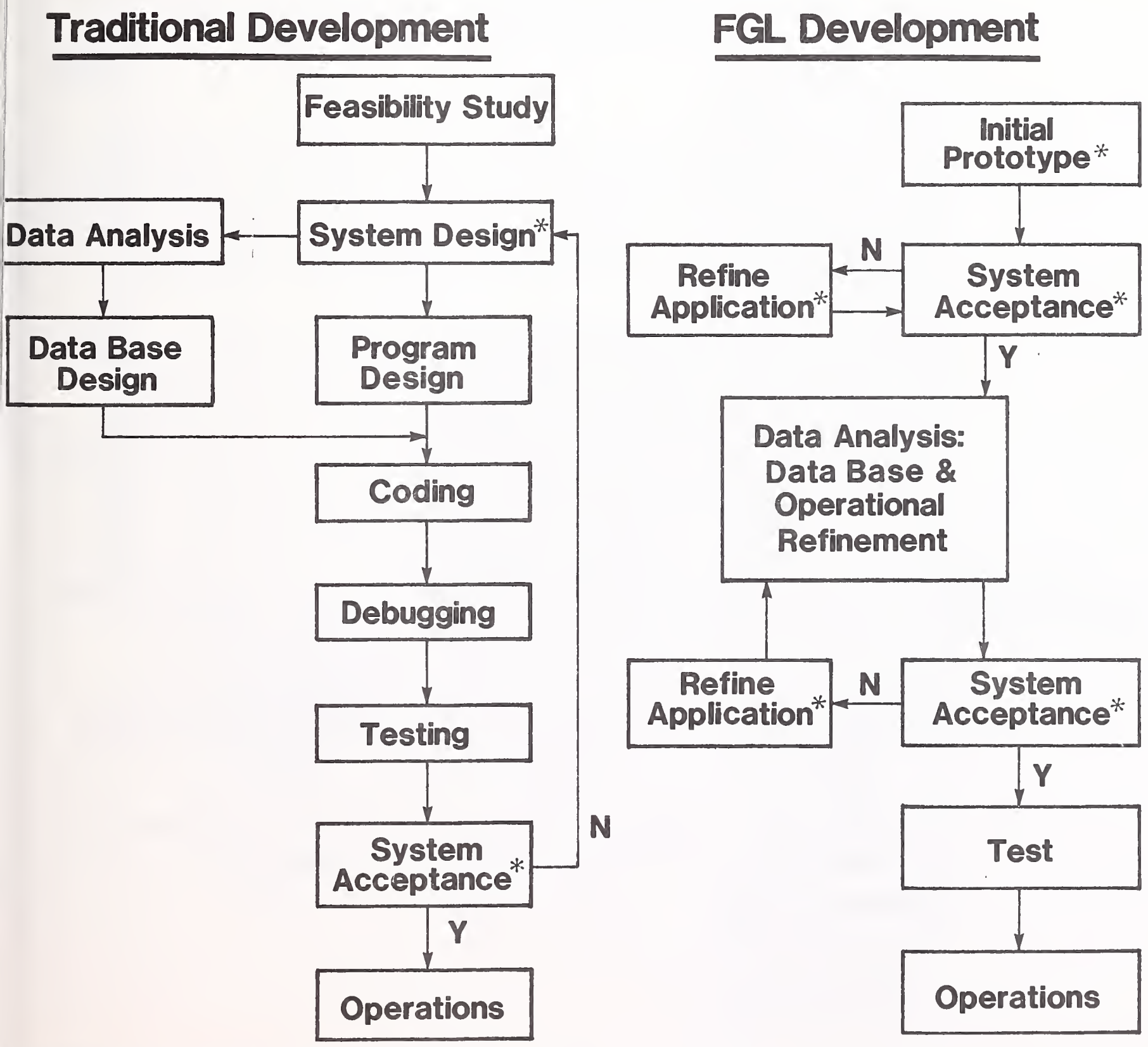
  - The speed and power of FGL development makes it possible to create a system prototype very quickly (i.e., often days versus months). Basic problems in approach or assumptions can be uncovered quickly.

  - Because an FGL can serve as a communications medium between IS and users, users can stay involved during the length of the system development process.

  - The "coding" process can concentrate on meeting user needs (i.e., logical processes) rather than correcting program and language syntax.

EXHIBIT II-6

FGLs CORRECT FLAWS IN TRADITIONAL SYSTEMS DEVELOPMENT



*Significant User Involvement

G. THE COSTS OF USING FGLs IN A PRODUCTION ENVIRONMENT CAN VARY GREATLY

- The current generation of FGLs essentially trade personnel time savings (80% is typical) for increases in machine requirements. Increased machine requirements are caused by a decrease in language efficiency: this varies widely, depending on circumstances, but the increase in required machine resources (largely CPU and memory) for a given amount of output can be conservatively estimated at 75% more than the COBOL equivalent; this should improve in the future.

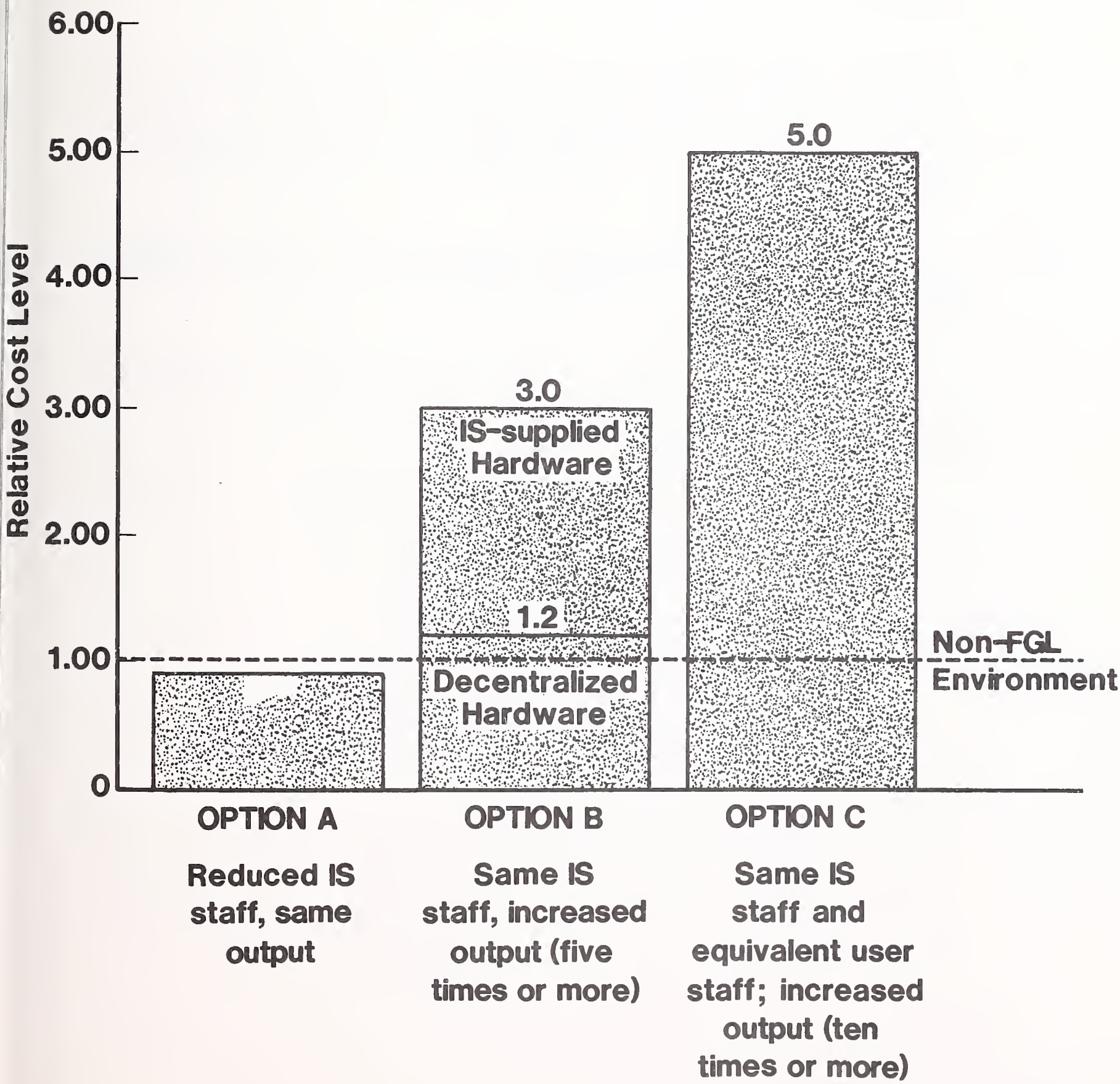
- The "A" option in Exhibit II-7 assumes an all-FGL environment, and that the level of development and maintenance output (i.e., application work accomplished) is the same as an equivalent non-FGL environment. Machine costs will go up, but not as fast as IS analyst and programmer costs would fall.

- Option "B," which is more realistic, assumes the pre-FGL staffing levels are kept. Consequently, many more applications can be produced and supported. Costs triple, assuming that all of the additional machine units come from the IS budget. There is a much more modest increase in the IS budget, however, if most of the additional hardware requirements are decentralized. Such a decentralization should not be a goal merely to diverting budget costs. It has the larger goal of being consistent with the logic of user control inherent in FGLs. It is also an efficient form of chargeback.

- Option "C" is the most realistic alternative since it assumes that user staff is able to make as large a contribution to development and maintenance as IS staff. While the IS personnel budget increases only modestly, machine requirements go up tremendously. These high machine resource increases are borne out by actual experience, e.g., CPU resources increasing by a factor of 10 over three years.

EXHIBIT II-7

THE COSTS OF USING FGLs IN A PRODUCTION ENVIRONMENT CAN VARY GREATLY



Percent cost level = 1.00

H. FGLs HAVE MORE POSITIVE THAN NEGATIVE FACTORS IMPELLING THEIR USE

- FGL machine resource demands will be balanced by steady improvements in language efficiency, and even faster improvements in hardware price/performance.

- Technical staff costs (and, to an extent, scarcity) and especially backlog problems will provide even stronger impetus to using FGLs in production situations. This is not to say that there are not negative perceptions of FGLs. IS fears include:

  - No (structured) methodology to replace system development methodology.

  - No firm divisions between system development steps, making it difficult to measure progress.

  - Obsolescence of IS technical and management skills.

  - Programmer opposition.

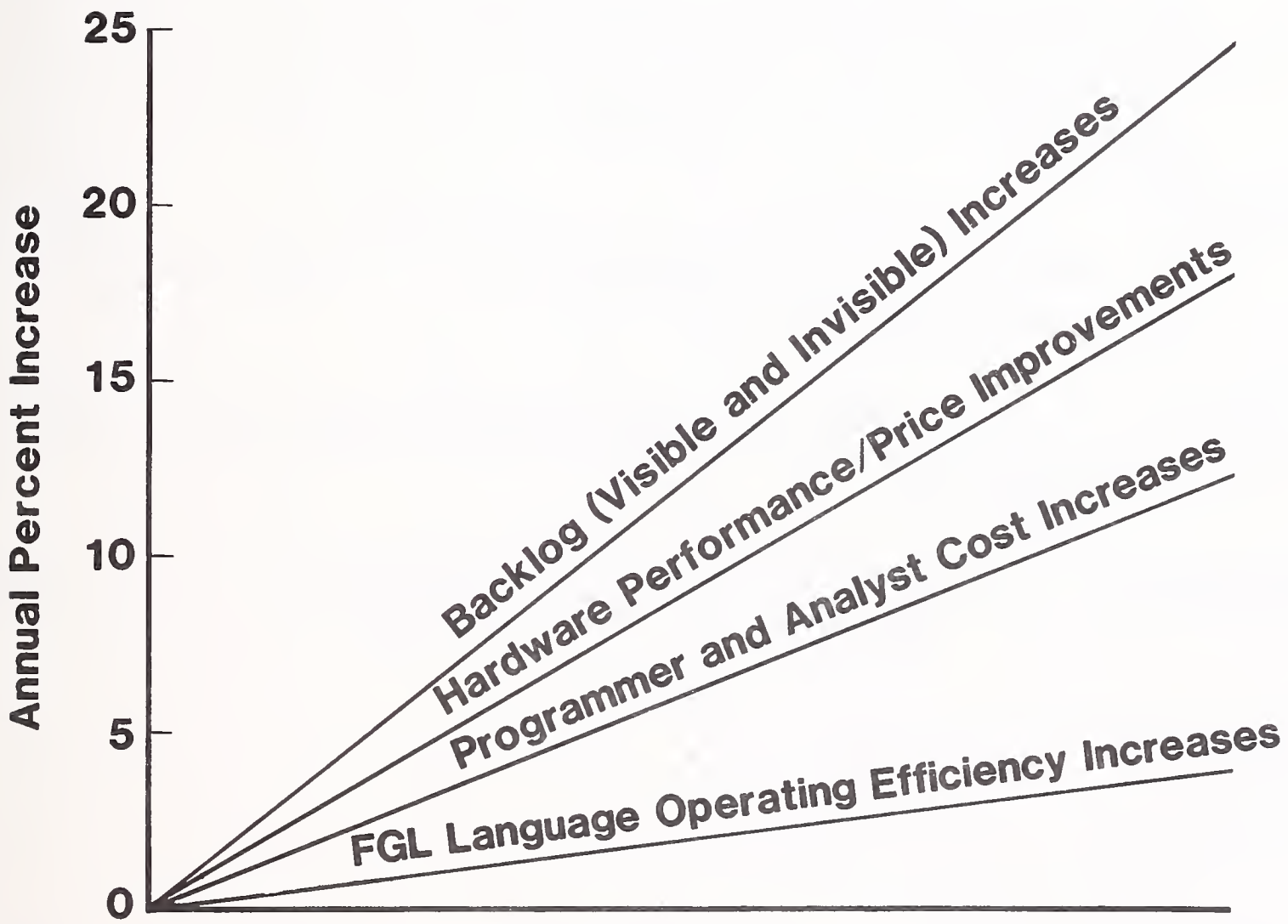
  - Loss of power and control by IS.

  - Marginal data processing applications may be implemented because of ease and/or lack of control.

- There is no question that development methodologies must be adjusted to take FGLs into account. The net result should be a significant improvement over current approaches. The other fears, while not invalid, are essentially management issues and can at least be neutralized by an FGL strategy.

EXHIBIT II-8

FGLs HAVE MORE POSITIVE THAN NEGATIVE FACTORS IMPELLING THEIR USE



I. FGL MAY HAVE A MAJOR IMPACT ON INFORMATION SYSTEMS STAFF

- FGLs will "de-skill" many "journeyman" COBOL programmers. FGL users with little experience will be able to outperform them.

- This will present IS with the opportunity to restructure, align, and integrate itself more closely with the rest of the organization.

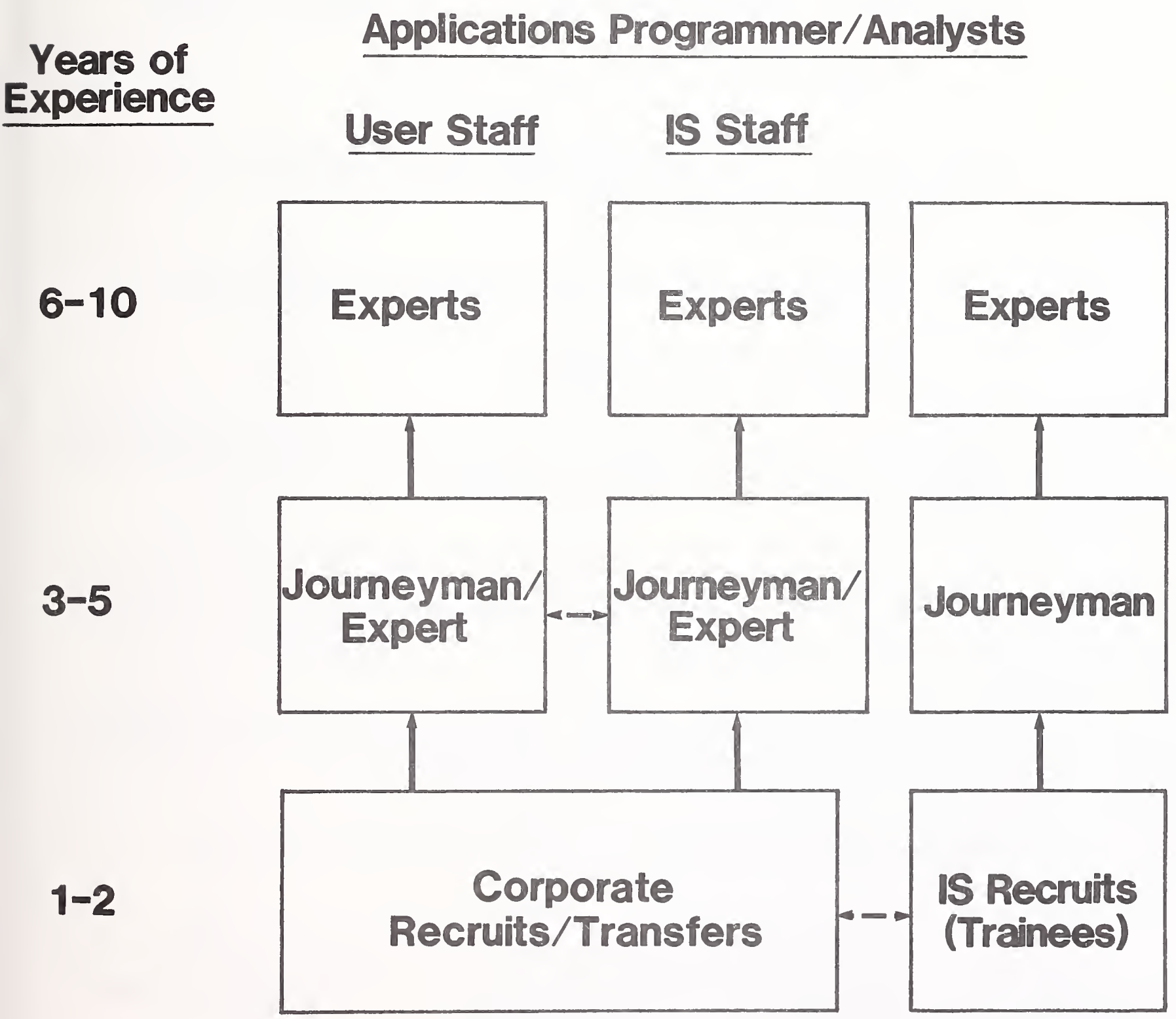
- This can, and should, extend to personnel policy and structure: application programmers and analysts would, except for experts, be a corporate resource, with a common resource pool, and personnel transfers between the IS and non-IS areas.

  - This will strengthen IS by giving it access to a broader and more diversified personnel pool.

  - It will also insert more computer system knowledge elsewhere into the organization.

EXHIBIT II-9

FGLs MAY HAVE A MAJOR IMPACT ON INFORMATION SYSTEMS STAFF

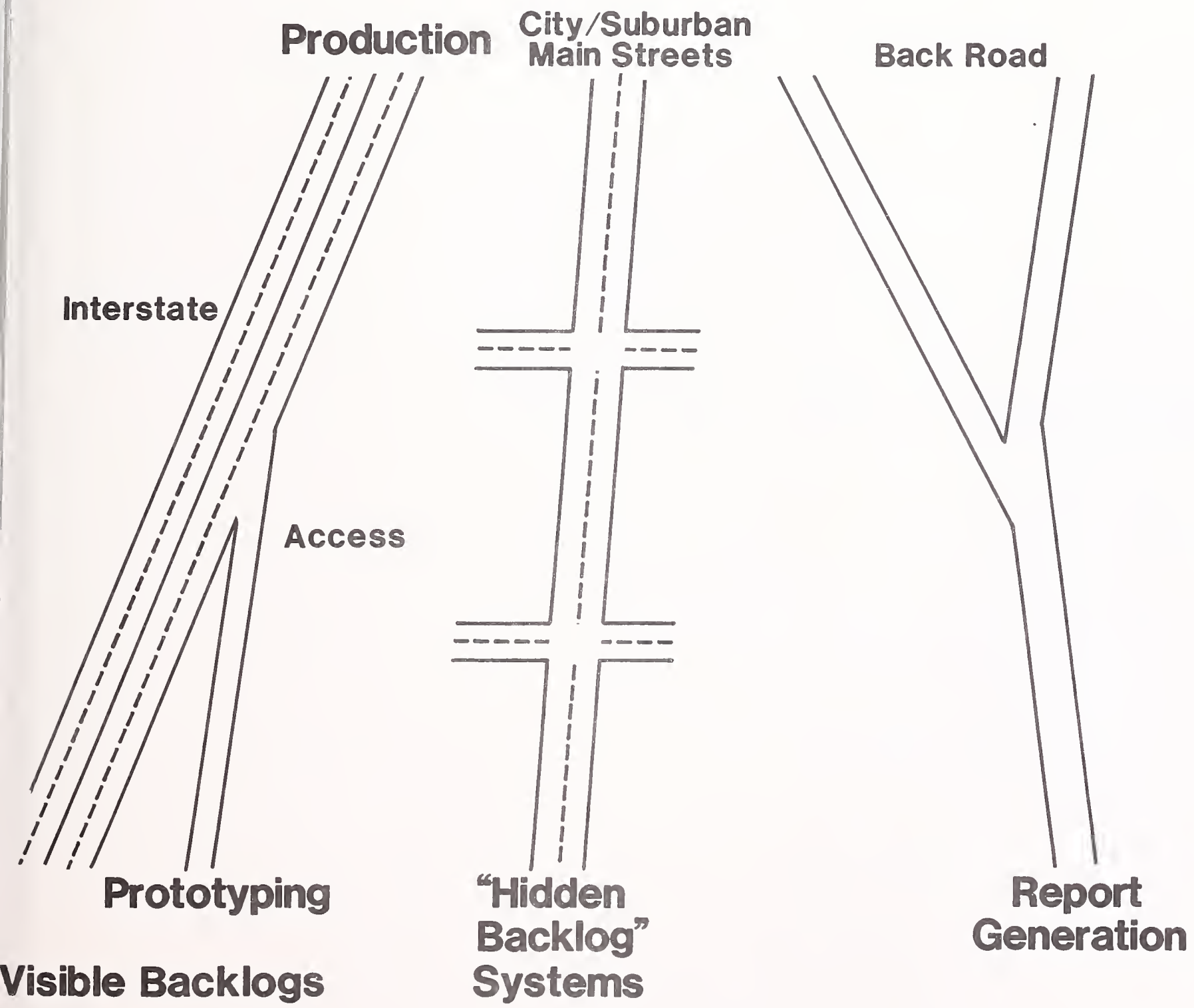


J. FOURTH GENERATION LANGUAGE PROVIDES ROUTES TO ACHIEVING INFORMATION SYSTEMS GOALS

- FGLs can directly support IS efforts to extricate itself from problems such as:
  - IS isolation from the organizational mainstream.
  - Creation of unsatisfactory systems, i.e., those that are too late, incomplete, or inflexible.
  - Perceived IS nonresponsiveness, especially concerning backlogs.
  - User ignorance of data processing.
- While FGLs used only by users (e.g., in the Information Center) can help, this approach will not get at the heart of the matter.
  - Under certain circumstances, heavy user FGL involvement could even make matters worse because users might believe that they had become more advanced than IS.
  - The bigger, and more likely problem is that users would be tantalized by FGL potential but be unable to realize it.
- However, FGLs are inherently attractive as IS-based production tools.
  - IS departments should introduce FGLs on a pilot basis, both to gain experience and to document costs and benefits in their particular environment.
  - It is most important to show what the impact on machine resources will be so that realistic budget requests can be developed and approved.

EXHIBIT II-10

FOURTH GENERATION LANGUAGES PROVIDE ROUTES TO ACHIEVING INFORMATION SYSTEMS GOALS



III WHAT IS A FOURTH GENERATION LANGUAGE?

III WHAT IS A FOURTH GENERATION LANGUAGE?

A. GENERAL DESCRIPTION

- An FGL has the following characteristics:
  - Nonprocedural (i.e., focuses on the result, rather than the process of obtaining the result).
  - English-like.
  - Nontechnical.
  - Flexible.
  - Fast initial learning period.
  - Often has built-in functions (e.g., DBMS, statistics, financial, graphics, text editor).
- COBOL is the foremost example of a third generation language. Exhibit II-2 contrasts third and fourth generation languages.
- There are at least four major types of FGL:

- Generalized tools.

- Tools linked to a specific proprietary DBMS.

- Application generators.

- Modeling languages.

- Exhibit III-1 gives examples of leading products in each category.

- The boundaries between these four types are already fuzzy and overlap considerably.

  - This overlap will become even more pronounced; for example:

    . Generalized tool FGLs have added modeling capabilities and links to specific proprietary data base management systems (DBMSs).

    . One FGL tied to a specific DBMS (NATURAL-ADABAS) has announced a link to the general IBM VSAM environment.

    . Modeling languages like EXPRESS and System W already have extensive user-friendly features.

  - Most vendors are already planning significant extensions and enhancements. (See Appendix B for details on specific vendor products.)

  - This blurring should become accentuated as vendors compete in offering complementary PC-based products. (See Chapter V.)

EXHIBIT III-1

TYPES OF FOURTH GENERATION LANGUAGE PRODUCTS

FGL	PRODUCT EXAMPLES
Generalized Tool	FOCUS INQUIRE RAMIS II NOMAD 2
Tools Linked to a Specific Data Base Product	NATURAL IDEAL On-line English
Application Generators	MARK V MANTIS
Modeling Languages	EXPRESS System W VisiCalc (Micro) 1-2-3 (Micro)

B. SELECTED FOURTH GENERATION LANGUAGE CHARACTERISTICS

- The definitions and classifications in the preceding section put the entire FGL phenomenon in perspective. However, the individual FGLs being offered are much more varied than the categories might imply.

- To give readers a sense of this variety, INPUT has provided sketches of selected FGLs. More detail is available in Appendix B.

1. FOCUS

- FOCUS is a highly integrated, general purpose FGL that can do just about anything the user wants, and is willing to pay for (in terms of added processing costs and additional modules). These characteristics include:

  - Information retrieval and reporting.
  - Transaction processing in a multiuser environment.
  - Direct interfacing to procedural languages and the most common DBMSs and operating systems.
  - Statistical anaysis.
  - Financial modeling.
  - Graphics.
  - Fully compatible PC version now available.

- FOCUS is a highly modularized system. The basic reporting module costs $43,000 ($1,500 on the PC), but a multifunctional system can reach the $150,000 range.

- The system has been aggressively and continually improved, both in performance and capabilities.

- A similar process appears likely for the PC version, which can be obtained and operated without the mainframe equivalent, at a slightly higher price.

- The PC strategy is to offer all the functionality of the mainframe version, subject only to size constraints in capacity on the smaller computer.

  - The PC version requires 256K of RAM, PC-DOS 2.0, at least one dual-sided, double-density floppy, a hard disk, and an accelerator board. It can use additional features if installed, such as more memory, a color monitor, or the 8087 math chip.

  - By fall of 1983, the system will provide distributed processing capability, and can function as an application prototyping and/or transaction processing workstation, as well as a training machine.

- FOCUS is highly user-oriented, but has been used successfully by some organizations as their exclusive programming and development language.

  - Operating performance ranges from 10% less than, to perhaps double that of a COBOL or PL/I program, but normally equivalent or 10% more if properly designed and tuned.

  - It may not be suitable for highly complex batch environments, but this would need to be determined on an individual basis.

- FOCUS offers extensive training and support in a variety of modes, including self-study, live instruction, computer-aided instruction (CAI) on the mainframe, and video-assisted instruction (VAI) from third parties.

2. NOMAD2

- NOMAD2 began as a timesharing tool from National CSS, was offered for a while as a turnkey product on their own midsized, plug-compatible mainframe, and is now available as a fully-licensed software product. A PC version is expected to be announced before the end of this year.

- NOMAD2 has extensive general-purpose capabilities within a single command language environment, and was originally used by National CSS to develop virtually all of their application systems internally, before being offered as a product.

  - For example, it supports the main Dun and Bradstreet credit reporting application, with over four million records in the data base, and 80 simultaneous users on the system.

  - It offers statistical analysis, financial modeling, graphics, and spread sheet capabilities within one set of commands.

  - The future PC version is expected to provide intelligence on both ends of the interactive loop, transforming data to and from native mode for several of the PC relational data base systems, such as 1-2-3, dBase II, Knowledgeman, and NPL.

- NOMAD2 costs about $125,000 or $135,000 with the financial modeling capability, including training and support. There are no other extra units or modules.

  - The product is exactly equivalent to the National CSS timesharing version, and presently operates under VM/CMS. A MVS/TSO version is scheduled for the first quarter of 1984.

- An extensive range of training possibilities exists, including a self-instruction book by Dan McCracken, on-site and vendor-site live instruction, and third-party video courses. Reflecting its timesharing origins, all documentation is machine-readable and accessible on-line.

- NOMAD2 has comprehensive security and integrity features that cover both data and functions. Security can be defined either vertically (application, file) or horizontally (item, value, user profile). The system uses shadow updating queued to a single manager to eliminate deadlocks.

- The strong financial modeling aspects of the system (FINAL) are a particular user view of the same system used by general information retrieval users, and have a consistent syntax with the rest of the system.

  - FINAL provides backward integration (i.e., "What must the factors be to produce this result?"), sensitivity analysis and capability to solve simultaneous equations.

  - These features are often included in standalone financial modeling systems, but not usually in the general purpose FGLs. The NOMAD2 user gets both.

3. RAMIS II

- RAMIS was the original FGL, even before anyone thought of the term "fourth generation language." It spawned, directly or indirectly, a number of the other leading FGLs, including its own successor, RAMIS II; further improvements are in store.

  - RAMIS II now offers the user-friendly front-end, Intellect, from Artificial Intelligence (also offered by IBM). RAMIS II calls it "ENGLISH," and provides some unique features not offered by Intellect's other implementors:

  - . A preloaded general lexicon that defines processing synonyms.

  - . The ability to switch back and forth between ENGLISH and native RAMIS II terminology.

- Processing inefficiencies associated with the original RAMIS have been significantly improved, and RAMIS II now claims to be the most efficient of the FGLs.

- A PC version based on the 68000 chip is being developed for release in late 1984.

- RAMIS II is a general purpose FGL that is intended as an information management product, but also offers interfaces to SAS for statistical analysis and graphing, as well as to the other common DBMSs such as IDMS, TOTAL, ADABAS and, of course, PL/I and IMS.

- RAMIS II is priced on a different structure than most of the other FGLs, and depends on the level of mainframe on which it is to operate. The prices range from $40,000 for the 4331 level, up to $80,000 for the 3081 level, for the basic system. Transaction processing and other modules are additional.

- RAMIS II has been used for some very large databases, including a million record market-research application. However, Mathematica, its vendor, does not sell prewritten applications.

- Extensive training options either exist or are being developed for RAMIS II, including live instruction at the vendor's or user's site, CAI (in the third quarter of 1983), and video courses from third-party vendors.

4. INQUIRE

- INQUIRE is a general purpose FGL that offers a unique feature to the user who works more with text than with numbers. It offers an extensive set of text retrieval and manipulation functions. INQUIRE can do automatic indexing, proximity searching, construct and maintain a thesaurus, produce keyword indexes.

  - While several of the other FGLs include a TEXT datatype, their editing and storage facilities are limited.

  - Conversely, although INQUIRE interfaces with SAS, it does not provide the range of statistical or financial functions provided by other packages. The basics are included, however.

- INQUIRE also includes a fully distributed relational database feature that can cross multiple CPUs so that the actual data location is transparent to the user. Most users, however, are not ready to utilize this feature.

- A wholly-owned subsidiary of INFODATA has been established to develop a version of INQUIRE for the PC. Specific product announcements will be made in October for first quarter 1984 delivery. The PC product will likely provide about 70% of the mainframe product capabilities.

- INQUIRE provides an integrated data dictionary that can include definitions of non-INQUIRE data items. It has a built-in retrieval path optimizer to provide high performance operational efficiency, and can be usefully employed by both programmers and end users.

- Statistical processing and graphics have been accommodated by a built-in interface to SAS. Comprehensive integrity, security, and auditability features are provided and controlled by the Data Definition Language.

5. ADABAS/NATURAL

- A number of the "classical" DBMSs have extended their facilities with powerful data manipulation commands and/or intelligent front ends, so that they are very similar to the general purpose FGLs.

  - One difference is the narrower breadth of functions supported.

  - Another is the smaller range of file interfaces offered.

- Among the leading representatives of the "extended DBMS" category is ADABAS with NATURAL. ADABAS is an inverted file DBMS, and NATURAL is Software AG's version of Intellect from Artificial Intelligence Corporation.

  - Like other representatives of this category, ADABAS provides ease of retrieving and manipulating information, but does not provide modeling or analytical functions.

  - There is a third party interface to SAS for statistical analysis.

  - ADABAS/NATURAL provides graphic display facilities in conjunction with IBM's graphics package GDDM.

- ADABAS is not inexpensive. The DOS/VSE version costs $106,000, and $172,000 for the MVS version DBMS, plus $40,000 to $60,000 for NATURAL, and an additional $15,000 for NATURAL/GRAPHICS.

- ADABAS is for programmers. It requires 10% of the coding effort of COBOL systems, 40% of the total development time, but furnishes operational performance at least as efficient as those same COBOL systems.

- NATURAL provides programmers and some end users a friendly, English language interface for information retrieval that is still under the control of the Data Base Administrator as to integrity, security, auditability, and performance. The relational nature of the underlying database provides flexibility without impeding performance.

- SUPER/NATURAL extends and simplifies the end user interface even more via a menu and QBE (Query By Example) approach.

  - SUPER/NATURAL enables secondary indices to be developed on-line, if not previously defined.

  - Little or no training is required to get at least some immediately usable results.

- Software AG is working on a PC version of ADABAS that should be announced by the end of 1983, and ready for delivery in 1984.

  - It will likely be a subset of ADABAS that retains the same syntax, but provides more limited capabilities, such as uploading locally prepared queries and downloading resulting data for further local processing.

  - The high performance structure of the ADABAS database is paid for in the storage requirements of the associated indexes.

- Nevertheless, in an office automation application the local indexing demands are not too great, while the adaptability requirement is exactly suited to the flexible inverted file structure.

  - Software AG expects to pursue the office automation approach, and exhibited a touch-screen version of ADABAS on a PC at their recent user conference.

6. ADS/ONLINE PLUS ONLINE ENGLISH PLUS INFORMATION DATABASE

- Cullinet was the first of the DBMS vendors to offer Artificial Intelligence Corporation's Intellect (discussed earlier in Section III) as an adjunct to their DBMS. OnLine English, as Cullinet calls it, is part of an integrated family of products, some number of which in combination can be considered to comprise an FGL. This section discusses OnLine English (OLE) plus Application Development System/OnLine (ADS/OL), an application generator.

- OnLine English works by querying an information retrieval file ("data file") and an index file of pointers into the data file. The data file may have been, but does not have to be, created and maintained by ADS/OL. The index file is normally created in advance, in batch mode by OLE, but can be created in realtime, with the next release of the product.

  - The query module accesses a lexicon of standard terms (such as who, when, how many, etc.) and user-defined terms (field names, synonyms, groups) to interpret requests and translate them into a computer-compatible format.

  - If a definition is not found, "near-misses" may be suggested by OLE or the user is given the opportunity to enter the definition on the spot.

- These features make OLE very user-friendly and easy to learn. But OLE also provides security, additional functions such as bar graphs, averages, and simple statistics including correlations, as well as auditability via a session logfile.

- ADS/OnLine complements OLE by enabling rapid development of transaction processing systems. ADS/OL automatically handles the details of:

  - Terminal input/output.

- Database housekeeping.

- Terminal screen formatting.

- Resource management.

- Data editing, decoding, and encoding.

- Error handling.

- The resulting code is almost 100% compiled, and provides performance that is the equal of COBOL or PL/I code. Procedures already written in these languages can be incorporated without being rewritten. As a result, development time is cut to approximately 10% of the COBOL or PL/I equivalent.

- ADS/OnLine costs $40,000 and requires IDMS, which costs $65,000 to $82,000. Both ADS/OL and OLE can use Cullinet's Integrated Data Dictionary, which costs $35,000. OnLine English costs $55,000, and can be used with or without IDMS files.

- Cullinet has also announced an arrangement with Micro Data Based Systems to offer The Knowledge Manager (Knowledgeman) as Cullinet's data base product for microcomputers.

  - Knowledgeman will extend IDMS facilities considerably by offering spread sheet, SQL-like query, statistical analysis, and structured programming capabilities.

  - Although all the details have not yet been resolved, the new product will enable data to be moved back and forth between micro and mainframe, while integrating a much broader variety of local display and processing functions.

- The new product will be available before the end of 1983, running on the IBM-PC under PC-DOS 2.0. It will be priced at approximately $1,000 per PC, but will also require IDMS-R (at $65,000) and Information Database (at $75,000) on the mainframe. Information Database is the link that extracts and translates data between IDMS, VSAM, or IMS files and Knowledgeman-accessible files.

7. MARK V

- Informatics General has updated its line of information handling products with Mark V, a generator for producing on-line application programs, similar to what Mark IV has been doing for batch programs for over fifteen years.

  - To the users of Mark IV, the new product will seem very familiar in concept. It fits naturally into an IMS environment and, in fact, uses native IMS facilities for READS and WRITES, as well as security, integrity and auditability.

  - A CICS version for DL/I and VASM files will be available in the next release in a few months. At that time, the user will have complete portability between the two systems by simply checking a single box on one of the application development screens.

- Informatics General intends this product for programmers, rather than end users. However, they have entered an agreement with VisiCorp to offer Informatics' Answer/DB product on the personal computer under VisiON.

- The Mark V product handles high volume transaction processing applications with at least the performance efficiency of COBOL, but at a development effort only one-fourth to one-half that of the COBOL effort.

- Mark V is priced at about $100,000 for the IMS/MVS version; the CICS version will be lower.

8. SYSTEM W

- Comshare, the timesharing service headquartered in Ann Arbor, Michigan, was the first to offer a distributed version of System W, their financial modeling package, on the PC, in January of 1983.

- System W is not a general purpose FGL, but given that many users limit their decision support activity to financial data, System W furnishes them extensive, easy-to-use facilities.

  - The system provides from four to nine "viewpoints," or dimensions, of data, that can be manipulated and reported in any manner or sequence the user desires. But the system incorporates enough intelligence to understand, for example, the valid sequences of consolidation and choose the correct one, even if the user entered something out of sequence.

  - The system operates on a set of rules for variables and time periods that are defined by the user in standard English syntax.

- System W has a large library of routines to handle forecasting, financial functions, goal-seeking (backward integration), sensitivity analysis, simultaneous equation resolution, and some graphics capability (using IBM's GDDM).

  - It operates on data that has been extracted from production files and/or data bases, and can produce journal files for updating back to those files, if desired.

- Distributed System W, the micro product, is 100% syntactically upward-compatible.

- It is distributed, not standalone, and depends on the security features of the IBM mainframe operating system at sign-on time.

- It operates under either CP/M-86 or PC-DOS 2.0, and requires at least 256K of memory and four dual-sided, double-density floppies or a hard disk to run.

- It is more limited than the mainframe product, however, in that it can only handle two views of data at a time.

- Comshare, as a computer services company, offers a wide variety of training and support facilities, including live instruction and videos, as well as a two-day CAI course on an Apple or IBM-PC. The latter has a one-time charge of $2,600 for the software.

- System W as a software product costs from $55,000 for a four-view capability, and up to $75,000 for the nine-view version. Distributed System W is $80,000, including two copies of the micro software. Additional micro copies are $995 each, but are discounted up to 50% in quantity.

9. SALVO

- Software Automation of Dallas, Texas, is now Beta testing for a September release of one of the first FGLs for microcomputers that does not start out as, or depend upon, a mainframe counterpart.

  - SALVO ("Software Automated Language Vocabulary") offers:

    . A relational DBMS providing a true "virtual JOIN" capability across, up to 16 files at one time.

    . A natural language interpreter that accesses a user-defined lexicon.

  - . An application generator independent of external host languages.

  - . Portability across CP/M and MS-DOS, as well as some specific micro vendor operating systems.

- SALVO is intended for the office professional, rather than either programmers or casual end users. It depends on the DP staff, however, for generating an extract of the organization's host data files or data base. SALVO then accesses these over a LAN or mainframe network.

- SALVO requires a minimum configuration of 64K user memory and two 30K floppies. Within this configuration it can handle up to 500 lines of data, a figure Software Automation claims to include almost all the FGL applications normally handled on a mainframe.

  - It provides security at field and file levels, accessed by user ID.

  - It will link to word processing, graphics, and spread sheet programs.

  - It is three to four times shorter and quicker to develop than other micro DBMSs, such as dBase II or MDBSIII, and probably 100 times shorter and quicker to develop than COBOL.

- SALVO will be marketed directly to Fortune 500 sized companies, as well as directly to computer vendors such as NCR and Hewlett-Packard.

  - The retail price is expected to be about $500 per copy.

  - Training materials include a user's manual, tutorial, and templates for common applications.

IV THE FOURTH GENERATION LANGUAGE ENVIRONMENT

IV THE FOURTH GENERATION LANGUAGE ENVIRONMENT

- This chapter examines current and potential uses of FGLs. The potential growth rates of use are also examined.

- This is a key element that both causes and affects FGL use. The economics of FGLs are analyzed in the second section.

A. FGL USE

- When looking at FGL use, the main question is: What kind of use is being emphasized?

  - Most current FGL uses can be typified as "secondary" uses; i.e., they can be very important, but are not affecting the mainline data processing system.

  - "Mainline" uses are those that introduce or modify a key system that is usually both large and transaction oriented.

1. SECONDARY FGL USES

- Secondary uses generally apply already processed data. These can be thought of as report generators or decision support systems. (While there is a distinc-

tion between these kinds of packages to IS professionals, there is considerable overlap in users' minds.)

- These secondary uses are classic Information Center applications.

  - Data files may be constructed especially for, as well as with the FGL software, or may be an extract from a production file. Many FGLs are able, for example, to act directly on IMS files.

  - In some cases a full production file may be acted on directly. However, many IS managers feel uncomfortable with users interacting directly with production files, even in a read-only mode.

- Another Information Center activity that is less visible because it does not meet the preconception of what an Information Center should be is the development by users of their own pseudo-production systems.

  - The management of one very large Information Center estimates that 10% of its considerable computing resources are being used for user-programmed and operated production systems.

  - In this context, a production system is one that is run on a regular basis, and processes data whose results are accessed by user management to further its business functions.

  - Often neither IS nor users are certain whether this is an appropriate activity, so assistance is only provided informally, with accurate statistics of use not readily available.

  - Many of these activities are conducted by users to reduce the "invisible backlog," i.e., the smaller jobs that users have been discouraged from formally submitting through the developmental request process.

- The hidden backlog's total resource requirement is probably as large as the visible backlog in most organizations.

- However, the individual components of the invisible backlog are much smaller than the jobs in the visible backlog. Consequently, they are very attractive candidates for FGL applications by users.

2. MAINLINE USES

- FGLs can be used in connection with a mainline application in two ways.
  - For prototyping production systems.
  - As production systems.
- FGL prototypes can be constructed quickly. Often alternative means of processing or outputs can be demonstrated. (Prototyping is discussed in more detail in Chapter V.)
  - In this approach, the actual implementation would be performed using a third generation approach (e.g., COBOL/IMS).
  - This approach is justified on the grounds that a third generation implementation is more efficient.
- The ultimate step, which only a few companies have made, is to use the FGL itself in a full-production environment.
  - For a variety of reasons to be discussed at length in Chapters V and VI, many IS organizations are not at all comfortable with this approach, whether IS uses the FGL alone or in conjunction with users.

- However, the alternative is that users will increasingly use FGLs independently and ultimately bypass IS.

- Exhibit IV-1 summarizes these mainline and secondary uses.

3. FGL GROWTH EXPECTATIONS

- Currently FGL use and growth is largely part of the Information Center issue. They are primary elements of the Information Center and will become even more important as FGLs become better understood and even more widely used.

- It is fair to say that an FGL is in use today in some part of virtually every company in the Fortune 500, as seen in Exhibit IV-2. However, currently, this use is fragmented.

  - Some FGL use is confined to isolated sections of a company, where, for example, the FGL is being delivered via a commercial timesharing service.

  - In other cases, FGLs are being used essentially as report generators in an Information Center environment.

  - As discussed in INPUT's report, Organizing The Information Center, (August 1983), a major defect in many current Information Centers is that adequate support is not provided. Consequently, most FGL users are unable to use the full potential of the FGL.

- The Information Center orientation will also predominate in the future, with nonprogrammer use expected to increase half again as fast as programmer use, as shown in Exhibit IV-3. FGL support is expected to lag far behind the growth in users. This is very similar to the situation that exists in Information Centers.

EXHIBIT IV-1

CLASSIFICATION OF FOURTH GENERATION LANGUAGE USE

Mainline Uses

- Fourth generation language prototyping, Third generation language implementation
- Evolved FGL prototype becomes production system
  - Users only
  - IS only
  - Users and IS jointly

Secondary Uses

- User report generation (typically, via Information Center) and analysis/forecasting
  - FGL files
  - Extracted product files
  - Production files
- "Hidden Backlog" satisfaction

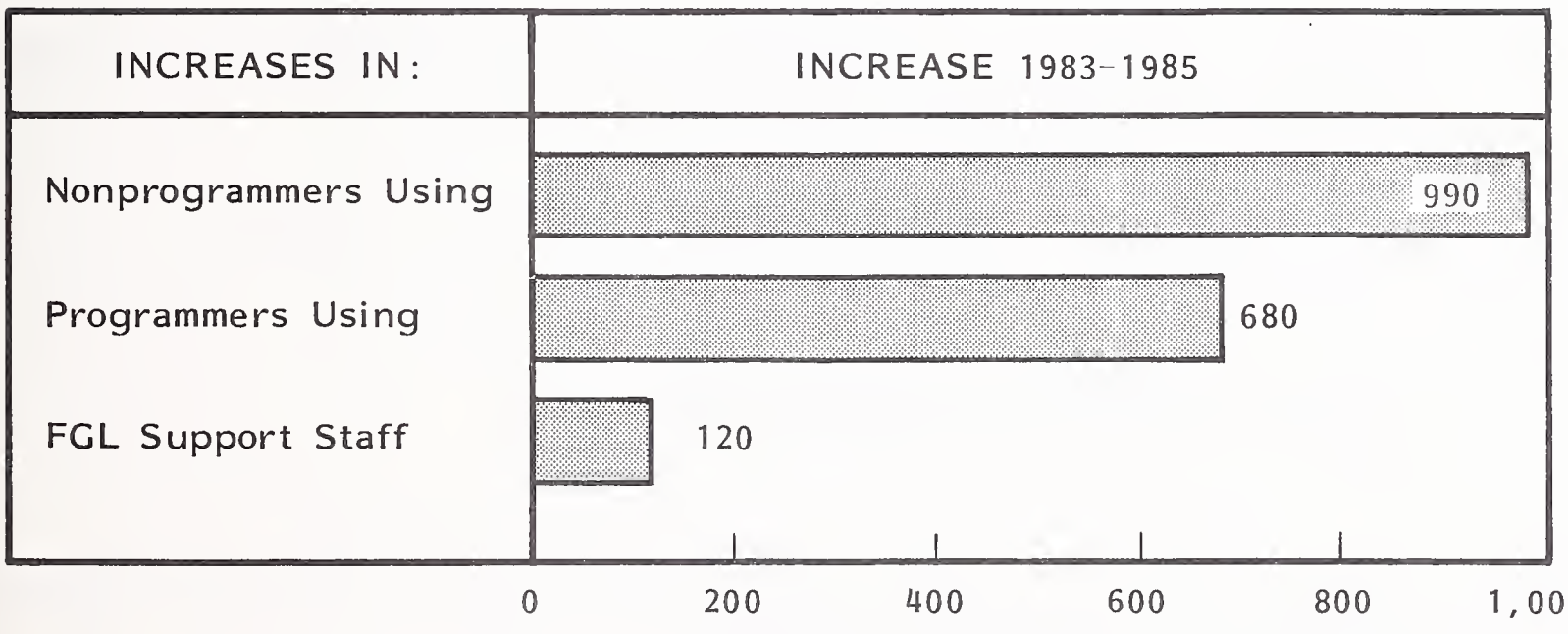
EXHIBIT IV-2

NUMBER OF INSTALLATIONS OF SELECTED FOURTH GENERATION LANGUAGE PRODUCTS

PRODUCT	NUMBER OF INSTALLATIONS			
	1980	1981	1982	TO DATE
FOCUS	150	275	525	705
NOMAD 2	7,000	10,800	11,900	15,000
RAMIS II	556	753	886	1,000
INQUIRE	N/A	N/A	N/A	300

EXHIBIT IV-3

EXPECTED INCREASES IN
FOURTH GENERATION LANGUAGE USE, 1983-1985



SOURCE: INPUT Survey

- FGL use by end users over the next several years should increase at the rate that resources are made available.

  - In the short run, IS can control the growth by making resources available via the Information Center.

  - In the longer run, FGL use will start to slip out of the control of IS.

- Actually, the biggest issue for IS will be the extent to which FGLs are to be used within IS and the use to which they will be put.

  - Much of the remainder of this report will focus on this issue.

B. FOURTH GENERATION LANGUAGE ECONOMICS

- Often, IS departments do not closely examine the economics of FGLs when obtaining them for Information Centers.

  - The Information Center users need tools like these, so their processing efficiencies are not critical.

  - Equally important, most FGL Information Center applications are not felt to be amenable to normal cost justification. Therefore, the economic costs and benefits of the delivery tool (i.e., the FGL) can also be deemphasized.

- The situation becomes quite different when examining the FGL's role in the IS department's "home turf" of mainline production systems. Here, the main objective issue becomes the economic question. (The more qualitative issues are discussed in Chapter VI.) The main economic issues are:

- To what extent are development and maintenance costs reduced when using FGLs?

- To what extent are production costs increased when using an FGL?

- The main cost factors are shown in Exhibit IV-4.

  - CPU and associated memory cost is the primary area of increased costs, conservatively estimated to increase by a factor of perhaps 75%. (Note that some vendors maintain that under the right circumstances efficiency may be comparable to that of COBOL systems.)

  - Personnel costs associated with system development or support are the main area of savings; personnel time is reduced by a factor of at least five.

  - Appendix C provides more details on FGL economics.

- The net cost impact is quite dependent on the way in which an IS department approaches an FGL implementation. There are four basic approaches:

  - Constant output, i.e., IS provides the same level of development and support output. This means a reduction in IS programming and analysis staff.

  - Constant inputs, i.e., the IS programming and analysis staff remains the same; processing requirements increase.

  - Constant input, decentralized output, i.e., as above, but processing requirements are largely decentralized (e.g., with personal computers).

EXHIBIT IV-4

FOURTH GENERATION LANGUAGE COST FACTORS

Machine Costs

- CPU/Memory
- Main Storage
- Storage Access

Development Costs

- IS Staff
- User Intermediaries
- End Users

Maintenance Costs

- IS Staff
- User Intermediaries
- End Users

- Users co-opted, i.e., IS programming and analysts work closely with, and are augmented by, user staff; processing requirements increase even faster.

- It should be noted that processing requirements can increase for these reasons:

  - The inefficiency of FGLs.

  - Increased system development activity (e.g., by utilizing IS staff more productively and/or mobilizing user personnel for system development activities).

- Exhibit IV-5 shows the cost impact if output is held constant: there is almost a direct tradeoff between personnel and hardware resources.

  - If IS staff is not reduced and IS must supply the additional processing resources, costs could increase by a factor of three, as shown in Exhibit IV-6; output would increase by a factor of five, however.

  - Total IS costs (although not, of course, total organizational costs) would not increase significantly if processing were decentralized. This approach would be attractive to many users (see INPUT's report Evaluating the EDP Level of Service, January 1982).

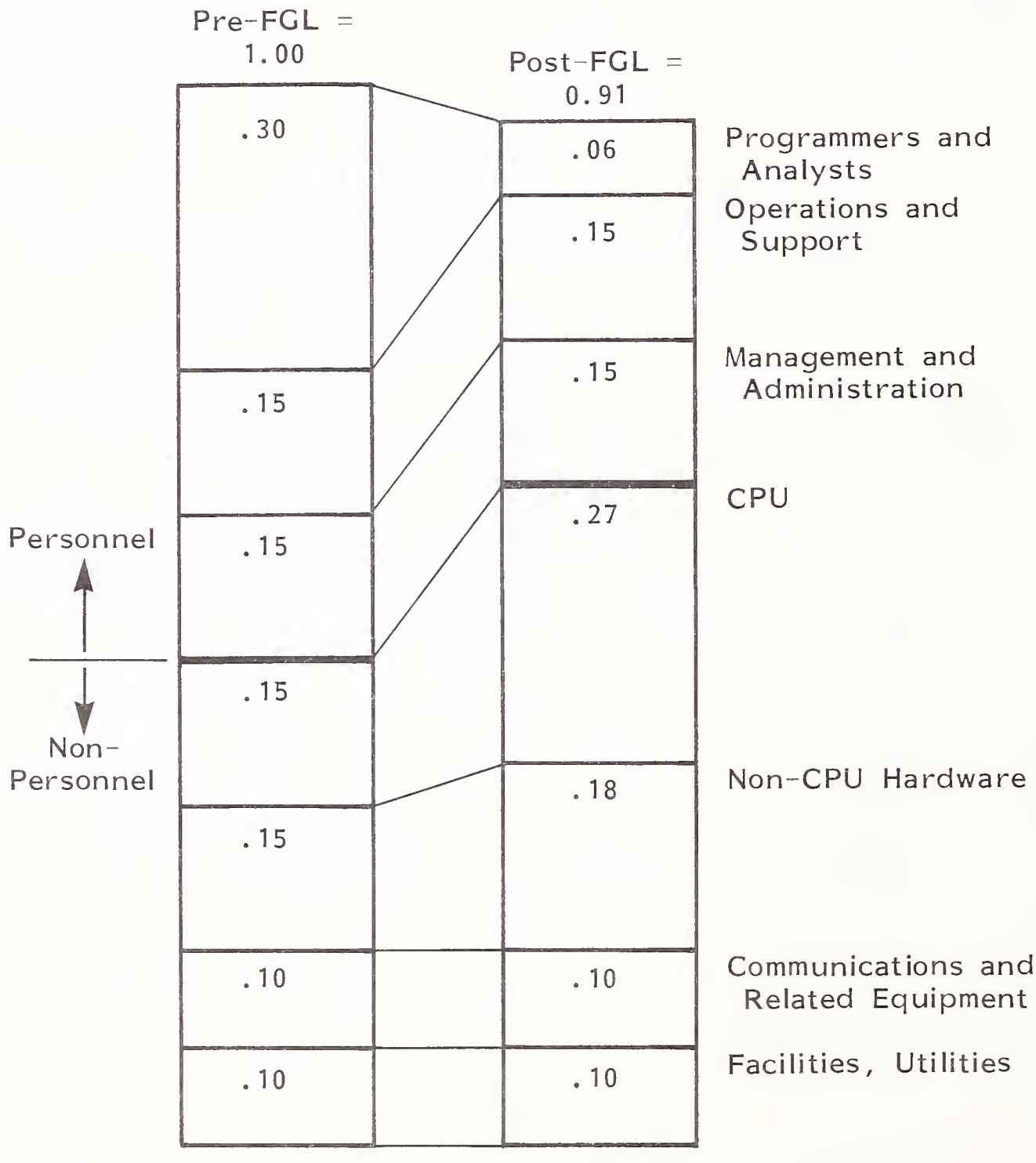
  - Finally, if user resources were mobilized to work in conjunction with IS, costs would increase by at least a factor of five, with output increasing by a factor of ten.

- The preceding is a model, of course. What happens in the real world?

  - Exhibit IV-7 tracks the installed millions of instructions per second (MIPS) of a medium-sized organization that decided in 1980 to use an FGL in virtually all of its systems.

EXHIBIT IV-5

POTENTIAL COST EFFECTS OF FOURTH GENERATION LANGUAGE USE IN A PRODUCTION ENVIRONMENT



SOURCE: Appendix C

EXHIBIT IV-6

COST IMPACT OF FOURTH GENERATION LANGUAGE USE AT DIFFERENT STAFFING LEVELS

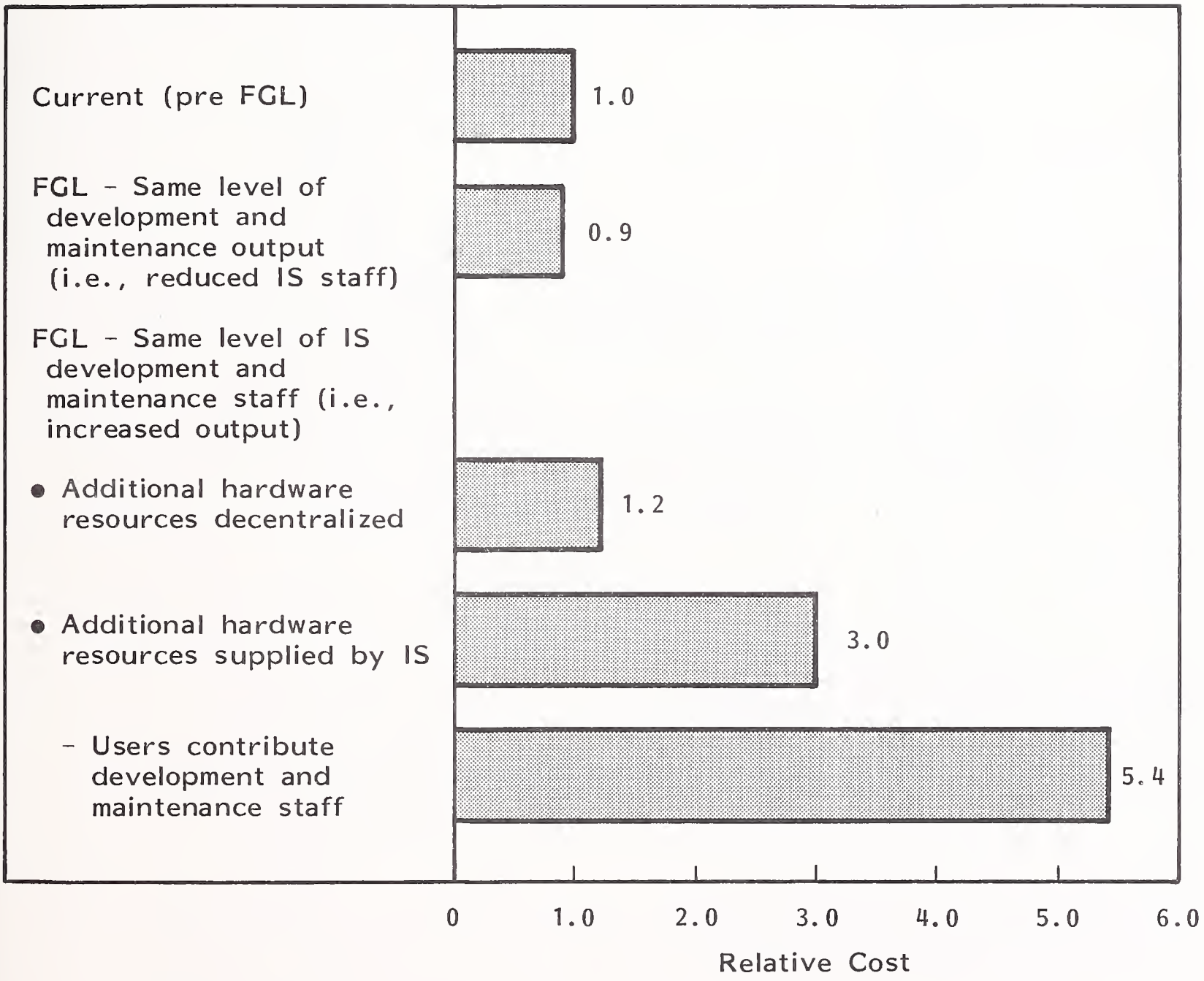
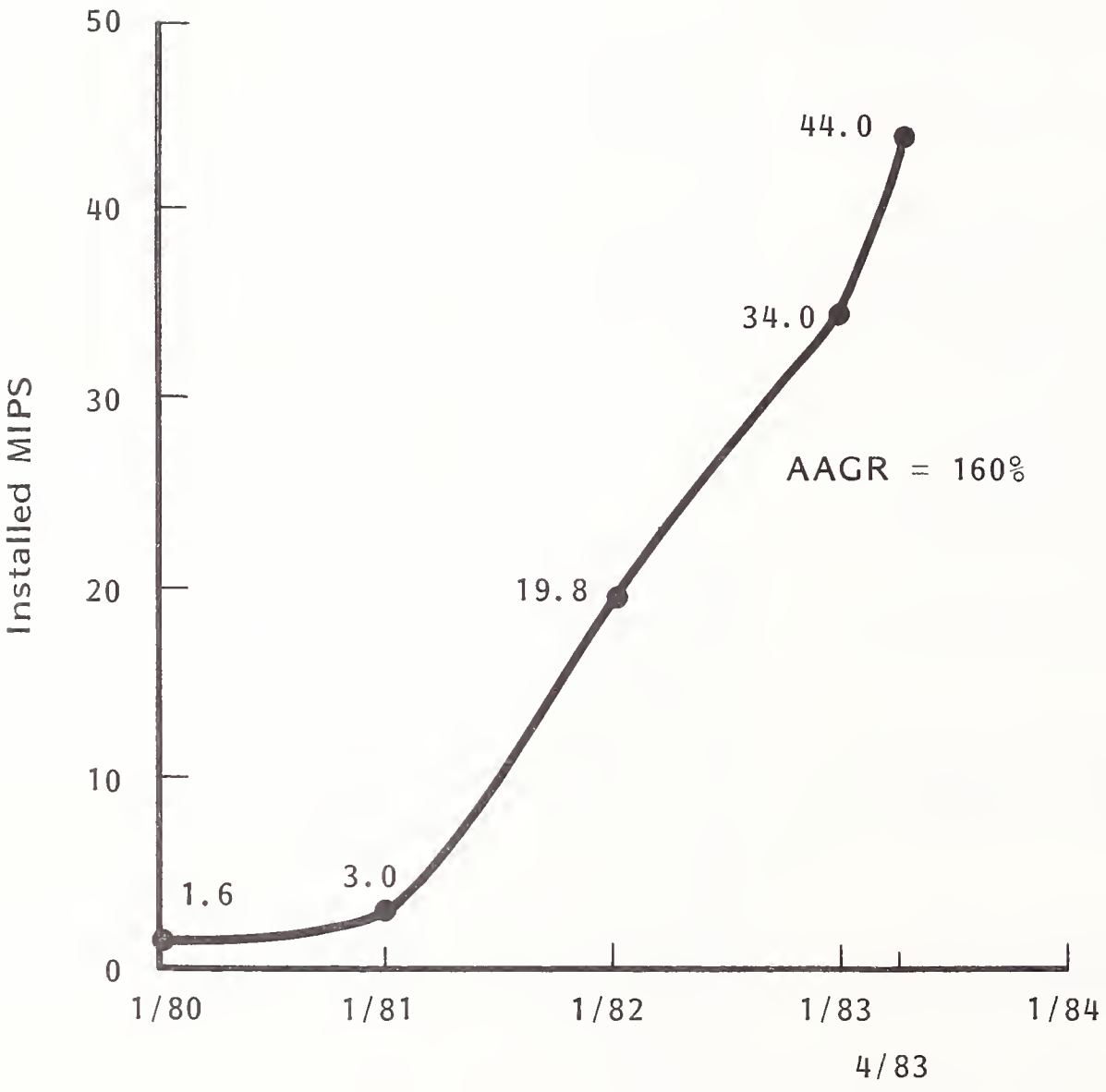


EXHIBIT IV-7

INSTALLED MIPS AFTER CONVERSION TO
FOURTH GENERATION LANGUAGE



  - . The MIPS increase is at an average annual rate of 160%, rising from under 2 MIPS to over 40 in little more than three years.

  - . MIPS to support development has grown by almost the same amount, as seen in Exhibit IV-8.

- This is felt to be an extremely successful operation. The user departments are fully involved and feel the investment has been well worth the time and energy.

- Many organizations would not view this kind of hardware increase with equanimity.

- It should be stressed that these economics are at a particular point in time. FGL economics should become increasingly more favorable to FGLs.

  - Hardware processing costs are continuing to fall, if anything, at a faster rate than previously, due to the advent of the super micro (e.g., a 1 MIPS machine sells for under $20,000).

  - Personnel costs continue to rise.

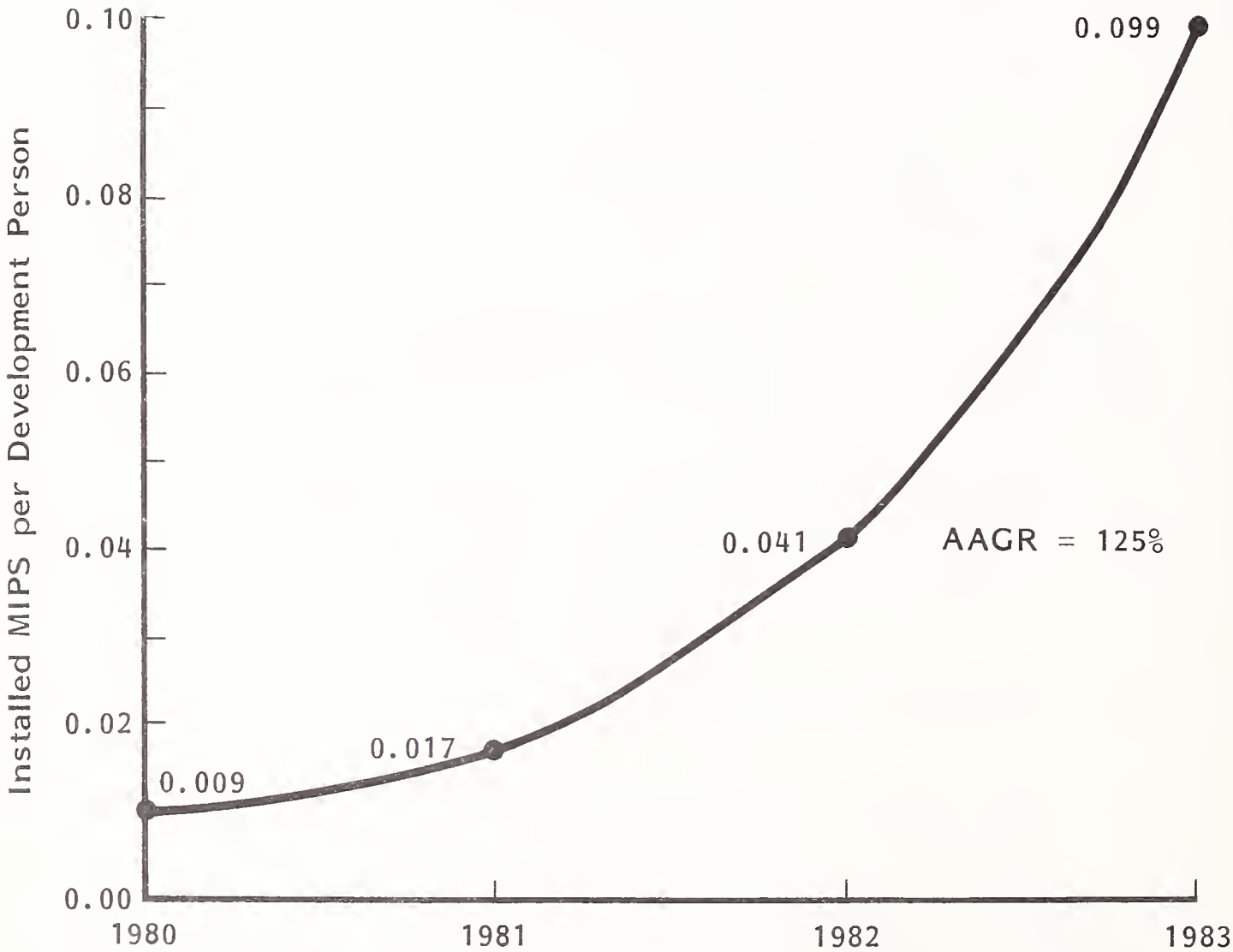
  - FGL vendors continue to work on efficiency; note the claims that code executes as efficiently as COBOL.

  - FGL "post-processors" or compilers may become more common.

- Critical pieces of FGL code can, of course, be rewritten into a more efficient language. However, as will be discussed later, this then undermines a considerable portion of the FGL advantage.

EXHIBIT IV-8

INSTALLED MIPS PER DEVELOPMENT PERSON
AFTER FOURTH GENERATION LANGUAGE INSTALLATION



- Limited tuning capabilities are already available in some FGLs (e.g., FOCUS, INQUIRE); these will doubtlessly be extended.

- Remember also that because of its efficiencies, assembly language was still being considered in the early 1970s as an alternative to COBOL. Also notice how FGL efficiency is being compared in terms of COBOL's. This, in itself, indicates how the decreasing price of processing is making language efficiency less important.

V THE USE OF FOURTH GENERATION LANGUAGES BY INFORMATION SYSTEMS DEPARTMENTS

V THE USE OF FOURTH GENERATION LANGUAGES BY INFORMATION SYSTEMS DEPARTMENTS

- As stated earlier, one of the most important issues that will be facing IS management is the extent to which IS itself will use FGL and in what ways.

- The typical current IS view on the use of FGLs is summarized graphically in Exhibit V-1.

  - As systems become larger and more transaction-oriented, FGLs become less likely to be used.

  - While it is true that an airline reservations system, for example, may not yet be a good candidate for an FGL implementation, even given the current state-of-the-art, there can be a good case made for FGLs in many classic transaction environments.

- Exhibit V-2 describes an actual, unresolved example where an FGL could provide a solution to a number of problems. It illustrates the difficulties some IS managers have in overcoming their image of the proper place for an FGL.

- There are four particular issues which bear on the applicability of FGL to help solve IS problems:

  - Prototyping with FGLs.

EXHIBIT V-1

IS VIEWS ON THE USE OF FOURTH GENERATION LANGUAGES

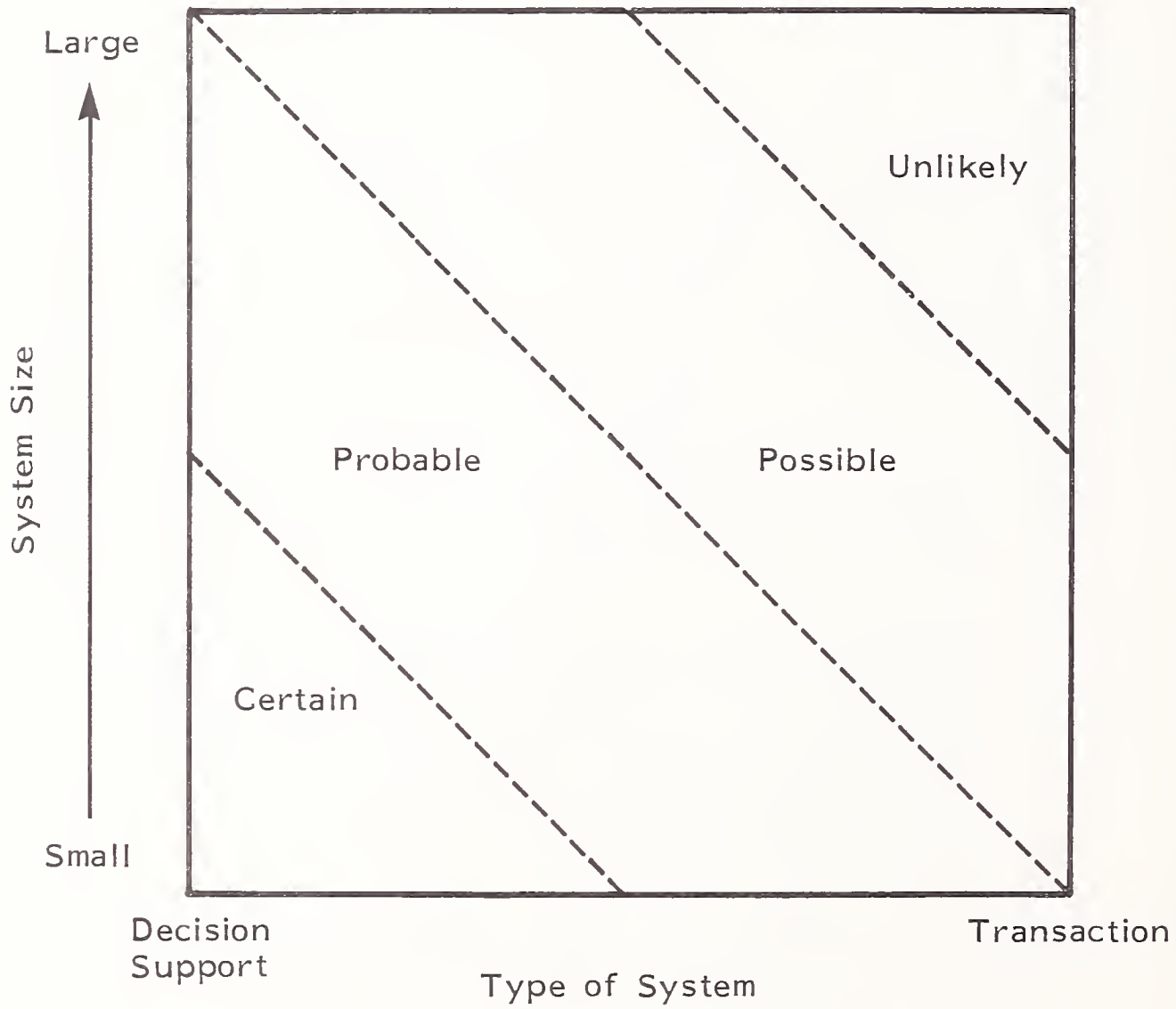


EXHIBIT V-2

A FOURTH GENERATION LANGUAGE
MANUFACTURING APPLICATION (CASE STUDY)

Environment:	• An equipment manufacturing division in a large, diversified corporation • A progressive data processing environment
Background:	• Four modules of a manufacturing software package were purchased several years ago at a cost in excess of $250,000 • Several million dollars were spent in trying to tailor the package to the division's manufacturing philosophy and interfacing the package with other divisional systems
Situation:	• The modifications have been unsuccessful, partly due to changing and unclear specifications • Several million dollars more would be required to meet current system specifications • Management no longer believes such projections • Management is insistent that the originally-promised 1984 date of completion be met
Potential FGL Solution	• Only a FGL implementation for the total system could meet the time requirement • The FGL would also be flexible enough to deal with further specification changes (such as changing manufacturing methodologies and philosophies). • Parts of IS management are convinced of the FGL feasibility; others are alarmed by the novelty

- FGLs in production systems.

- FGL as a mainframe-PC link.

- FGL's impact on quality.

A. PROTOTYPING

- It is increasingly recognized that the system life cycle approach, to which so much effort has been devoted over the past decade, is not completely on target, as illustrated in Exhibit V-3.

  - It is not so much that the goals of the life cycle approach are wrong, it is more that it assumes:

    . A "correct" solution can be defined with very limited (often no) iteration.

    . Once found, the correct solution will remain stable for a long time.

    . IS staff and users can communicate effectively.

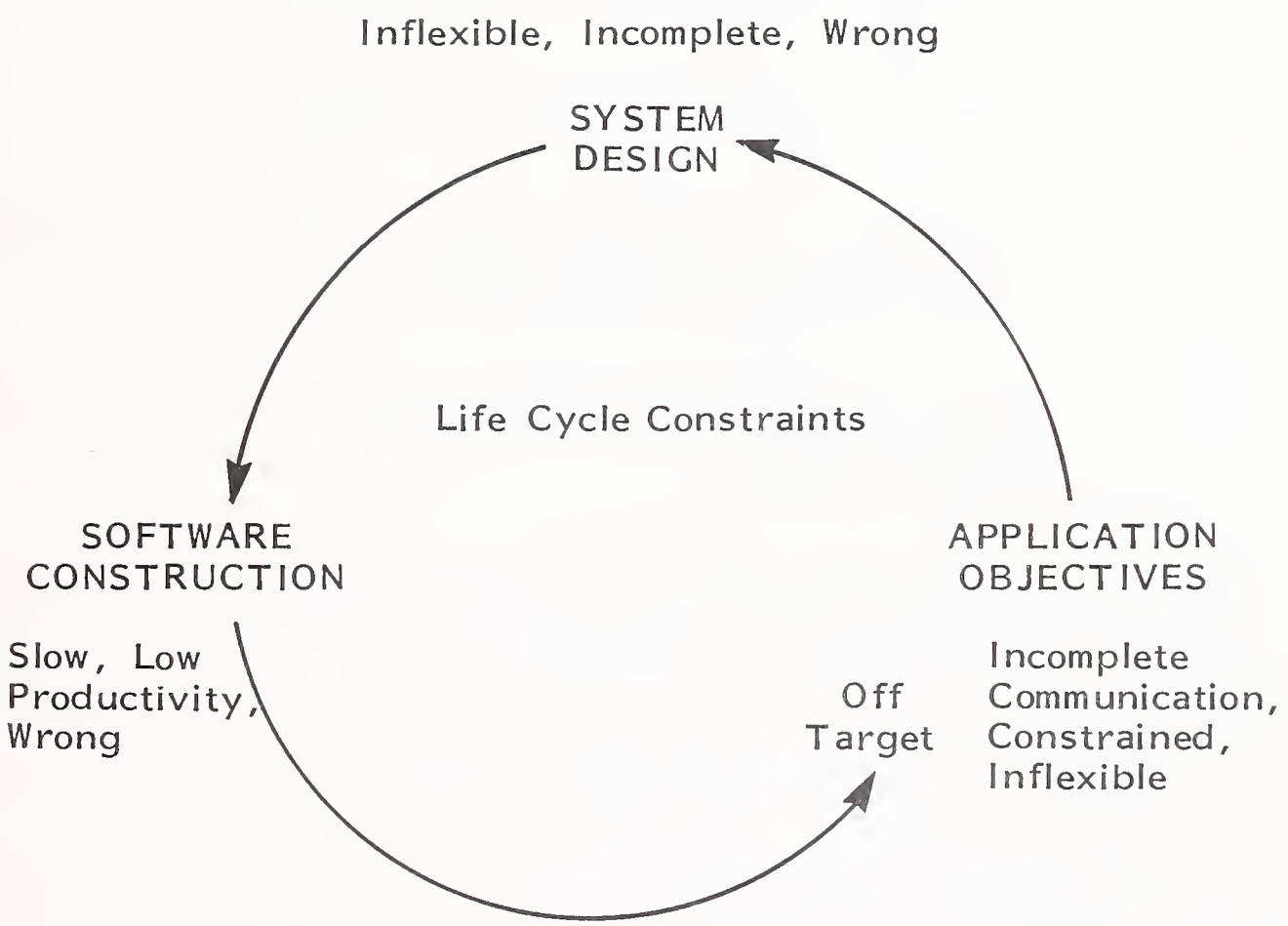
- In fact, what often happens is users sign off (or, worse, delegate the signing off) on a very large stack of documents that they barely understand. After months or years, their reaction to the finished product is all too often:

  - "That's what the previous management wanted."

  - "That's very nice, but what I really wanted was . . . ."

EXHIBIT V-3

THE VICIOUS CIRCLE OF TRADITIONAL SOFTWARE DEVELOPMENT



- "You didn't understand."

- There are good data-driven system design methodologies that are as useful with FGLs as with third generation solutions. However, even the best methodology is now dependent on systems analysts, generally both user analysts and IS analysts, as seen in Exhibit V-4.

  - Every line shown on Exhibit V-4 is a potential weak join.

    . The users do not think in "data" or in "processing" terms.

    . The user and IS analysts' tasks are often poorly defined and they often end up in competition rather than cooperation.

    . The analysis becomes a formal exercise not solidly grounded in realistic needs.

- These same problems have existed in engineering disciplines since their origin. The prototype was soon developed to:

  - Prove to the engineer that the solution would work.

  - Confirm that the solution was to the correct problem.

- Prototypes have rarely been practiced in a third generation environment. FGLs are uniquely suited to create working prototypes:

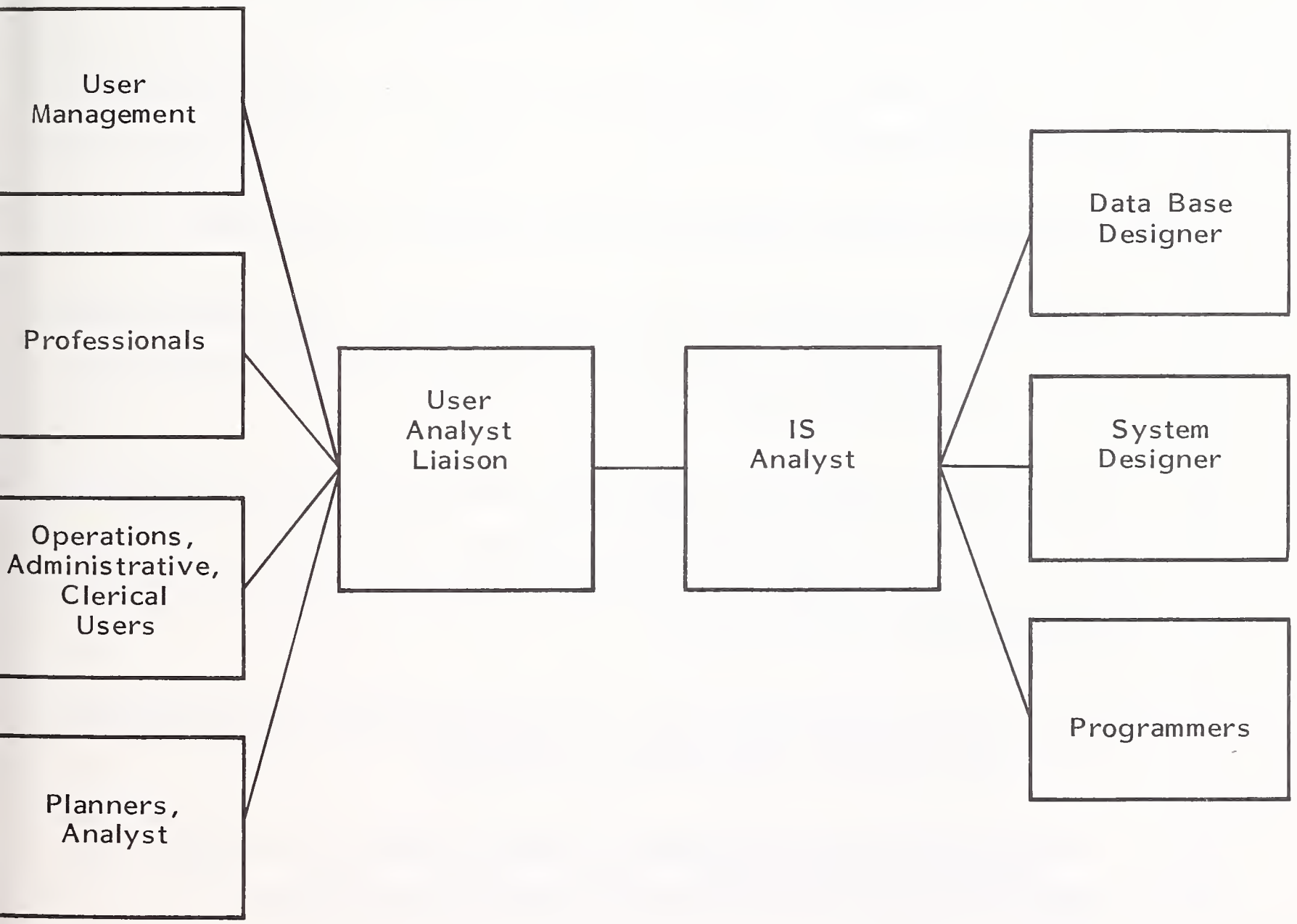
  - Prototype creation is rapid.

  - End users can participate, in a hands-on mode if desired.

  - Adjustments and alternative solutions can be quickly presented.

EXHIBIT V-4

ANALYSTS - WEAK LINKS AND BOTTLENECKS?



- If desired, the prototype itself can be scaled and used as the production system.

- The prototyping process can rectify many, if not all, of the shortcomings of the life cycle approach. It can also make the analysis process become real to the typical end user.

  - Rather than, at best, dummy reports being created, real reports, based on real data, can be presented within a few days of a work session.

  - Besides the obvious credibility that this produces, the analysis process can proceed at a pace much closer to human thought.

  - The process is self-documenting, in a way that is accessible to both IS and user staff.

- This is not to say that there are not drawbacks to FGL prototyping.

  - The chief problem is that the process is new. There are only a few methodologies now available for driving the prototyping process (e.g., DACOM's POM-80).

  - More basically, there will probably never be a completely satisfying set of formal rules to describe the desirable iterative process. Two iterations will be adequate in some circumstances and then, may not be in others.

  - The very acceptance of FGL prototyping may lull IS staff into believing that little or no formal testing is required.

- However, even with the potential problems, prototyping can have a high impact on most system activity areas, as Exhibit V-5 illustrates.

EXHIBIT V-5

PROTOTYPING EFFECTS ON SYSTEM ACTIVITIES

ACTIVITITY	PROTOTYPING IMPACT	TYPES OF EFFECTS
System Feasibility	Very High	Can replace paper execises.
System Modeling	High	Alternative systems can be tested.
System Design	High*	Can lay out working model of system quickly.
Program Design	High*	Can lay out working model of program quickly.
Coding	High	Multiples of efficiency.
Debugging	Medium**	Easier to identify logical errors, if tested.
Operations	High	Negative impact; significantly more computing resources required.
Maintenance-error correction	Medium**	Code much more self-documenting.
Maintenance-enhancements	Very High*	Recapitulates benefits from "Feasibility" to "Coding".

* Only if prototype moved into production, otherwise medium impact.
** Only if prototype moved into production, otherwise low impact.

- Note that the positive impact of prototyping on system design, program design, and maintenance is lowered if some or all of the production system is programmed in a third generation language. This kind of hybrid third and fourth generation system might otherwise be favored on the grounds of hardware/production efficiency.

  . One cannot be positive that the correct prototype logic will have been carried forward into the final system.

  . Debugging and maintenance will be against third generation code. At best, the FGL code will be a form of documentation (that may not be kept up to date).

- The high (and desirable) amount of changes to FGL prototypes will mean a higher demand on computer operations (e.g., see Exhibit IV-7).

B. FGLs IN PRODUCTION

- As noted in the previous section, a number of the benefits from using prototypes to increase accuracy are lost if the prototyped system is not carried forward into production.

  - In addition, many of the speed and productivity benefits are lost since the FGL code must be reprogrammed as third generation code.

  - This approach can only be recommended if there is a serious, documented case of potential hardware inefficiency and the dollar tradeoff prevents taking the FGL approach.

    . Users will often prefer, and pay for, the seemingly more expensive option that delivers a more acceptable system more quickly.

. This has been repeatedly proven by the success that timesharing companies have had in selling their expensive solutions.

- One of the biggest benefits of implementing an FGL production system is that it then becomes feasible to have users directly maintain the system. It may also be feasible for the user department to run the system on a decentralized basis.

  - By removing IS from the routine support business, IS can begin to extricate itself from the unrewarding maintenance business. (See INPUT's November 1982 report, Software Maintenance, The Uninvited Guest.)

  - IS will still have to provide technical backup and assistance. However, user departments can make the change to logical construction, which they are by far the most qualified to carry out.

- Much of the discussion up to this point has been implicitly assuming the implementaton of a new system that has not been computerized before. Such systems are quite scarce. Most "new" systems are in reality reworkings of previously computerized systems.

  - Such system conversions are at least as feasible for FGL as are completely new systems.

  - The most important requirements are that there be a defined system owner willing to assume responsibility, and that user needs are volatile.

  - These and other somewhat less important factors are described in Exhibit V-6.

EXHIBIT V-6

FOURTH GENERATION LANGUAGE CONVERSION TARGETS

FACTOR IN FGL CONVERSION	IMPORTANCE
User willing to assume responsibility	H
Constantly changing user needs	H
Defined user system ownership	H
Old system	M
Difficult to maintain	M
Error-prone system	M
High maintenance costs	M
High level of complaints	M

H = High
M = Medium

- The most critical issue when considering using an FGL in a production setting is whether to use an FGL or an application software package. The most important advantages of a software package compared to third generation custom programming are:

  - Speed of implementation.

  - Cost (i.e., 5%-20% of the cost of internal development).

  - Accuracy and integrity.

  - Vendor maintenance.

- These advantages have helped fuel an annual growth of over 30% in the packaged software industry for a number of years.

- However, these advantages only exist in a pure form where modifications to the underlying package are few or nonexistent. This means that many businesses are faced with two unattractive alternatives when making a packaged software purchasing decision.

  - The package is bought and installed as is; often, significant portions of the company's business operations must be changed to conform to the package - the tail wagging the dog. By doing so, the company must sometimes give up unique, valuable characteristics and may also find it difficult to have its systems react flexibly to outside events.

  - On the other hand, extensive package modifications can add greatly to costs and elapsed time, and often the changes are so extensive as to mean that vendor maintenance is no longer viable. As the example in Exhibit V-2 shows, sometimes such attempted modifications are not even feasible.

- Recognizing these issues, some companies are now buying application packages merely to use as a "shell" to write a custom application around. They find that this approach can save them a year's elapsed time and a modest amount of development cost (up to 25%). They willingly give up most of the benefits of packaged software in order to secure a solid software foundation and to save time.

- Using an FGL can be yet another option to choose from when planning major software projects.

  - Many of the speed and cost advantages of acquiring a software package can be gained by using an FGL for implementation.

  - The accuracy and solidity of an FGL can be as high as that obtained by using a software package (assuming that testing is adequate).

  - More computer resources may be necessary to run an FGL system; on the other hand, generalized software packages often have high overheads, including support for many functions that a particular customer has no need for--at least in the form offered.

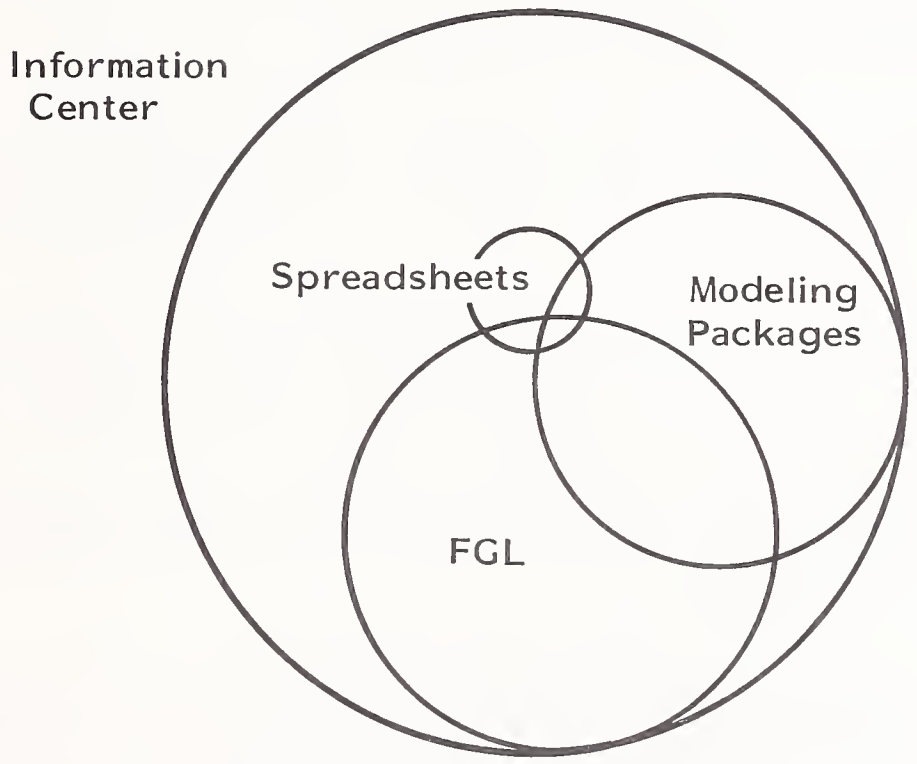
C. MAINFRAME PERSONAL COMPUTER FGL LINKAGE

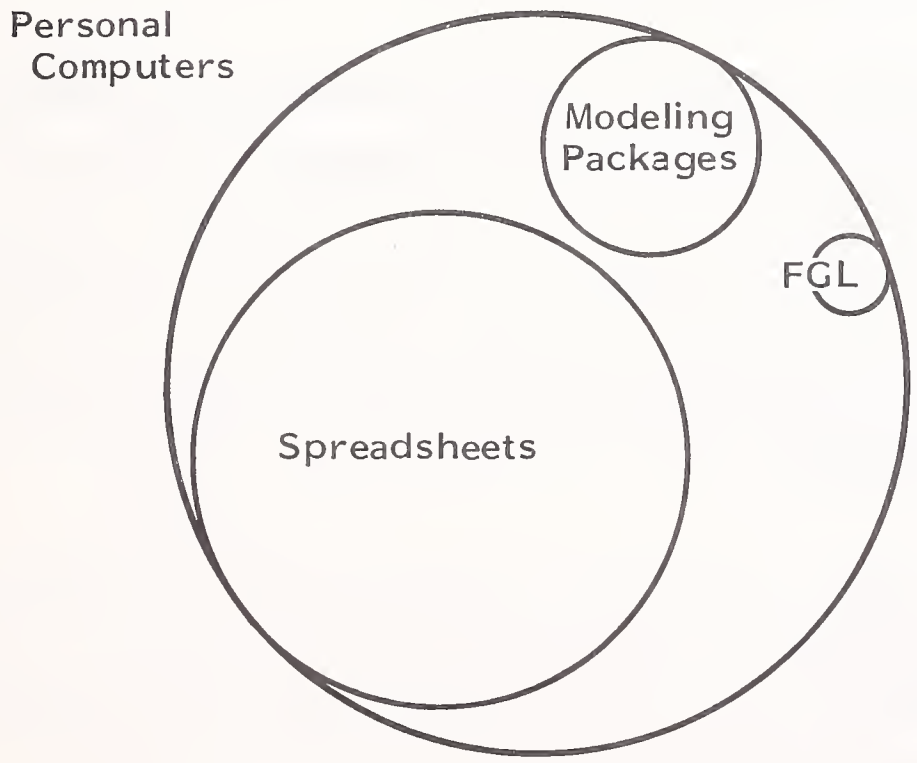
- Currently, mainframe and PC systems are essentially isolated as seen in Exhibit V-7. PCs may be used as terminals and/or extracted files downloaded, but from a processing and application standpoint, there is little connection.

- This is in the process of changing, thanks to the initiative of many of the major FGL vendors.

EXHIBIT V-7

INFORMATION CENTER AND PERSONAL COMPUTER ISOLATION





- FOCUS, for example, has recently released its first report writing module that is planned to lead to an almost complete transfer of its mainframe-based capabilities to PC in the course of the next year. System W from Comshare is in a PC version that contains almost all the features of its mainframe product.

- FOCUS and System W are not alone. Many of the other leading FGLs (e.g., NOMAD, NATURAL, RAMIS) plan similar transfers to PCs.

- Competitive pressures will force most, if not all, FGLs to offer complementary mainframe-PC products in the next two years.

- It is still not clear what precise relationships will be established between mainframe and PC products.

  - Most vendors will be attempting to have the PC possess the same functionality as the original mainframe version or, at the least, be a comprehensive subset.

  - Vendors will try hard for functionally identical versions so that both uploading and downloading will be transparent to users.

  - Where there is such bidirectional compatibility, it will be possible to prepare FGL programs on either the mainframe or PC and run them on either one. Disk storage capacity will be an obvious limitation for PCs, at least initially.

- To compete in the PC marketplace, mainframe-based FGLs may have to offer new or expanded features such as spread sheets, graphics, etc. These features may then get fed back into the mainframe product, both to preserve compatibility and to work as a marketing tool.

- EXPRESS is taking another approach. They expect to have a turnkey super-micro-based version of their system available in 1984 in the $20,000 to $25,000 region.

  - Given EXPRESS' power and complexity, it is understandable that they would take this approach to offer a significant subset of their mainframe product's functionality on a standalone basis.

  - It is quite possible that the EXPRESS decision signals a branching of FGL PC-based products.

    . For routine, small-scale, or slow response applications, medium-power PC (e.g., the IBM-PC) will be quite adequate.

    . On the other hand, high-priority, large-scale, heavy-computational or fast-response applications may need more high-powered machines (e.g., 68,000-based machines like the Charles River Systems or Stratus hardware).

- The net result of the expansion of FGLs to PCs will be a linking of PCs and mainframes on a software and application basis; this will encourage further hardware integration and development of distributed processing, as seen in Exhibit II-5.

  - The high-powered micros may become the data base machines of the future. Current data base machines tend to be "backend" rather than "frontend."

  - Personal-computer-based FGLs may resolve the dilemma of resource drain on the mainframe caused by FGLs, yet provide the information-center-like tools that users crave.

- As the exhibit shows, the tightened links between PCs and mainframes will make it easier to develop FGL-based production systems off-line. The mainframe will not have to absorb the sudden, high-priority peaks of development resource requirements.

D. FGLs AND SYSTEM QUALITY

- Increasingly, IS departments are becoming concerned about system quality. It has become clear that system quality is the foundation for both productivity (as Exhibit V-8 illustrates), user satisfaction, and life cycle maintenance costs. This is an important motivation for IS departments setting up quality-assurance programs and units.

- Three of the most important constituents of quality systems are:

  - System robustness.

  - Flexibility.

  - Data integrity.

- FGLs can directly support each of these constituents.

  - FGL-based systems can be robust due to:

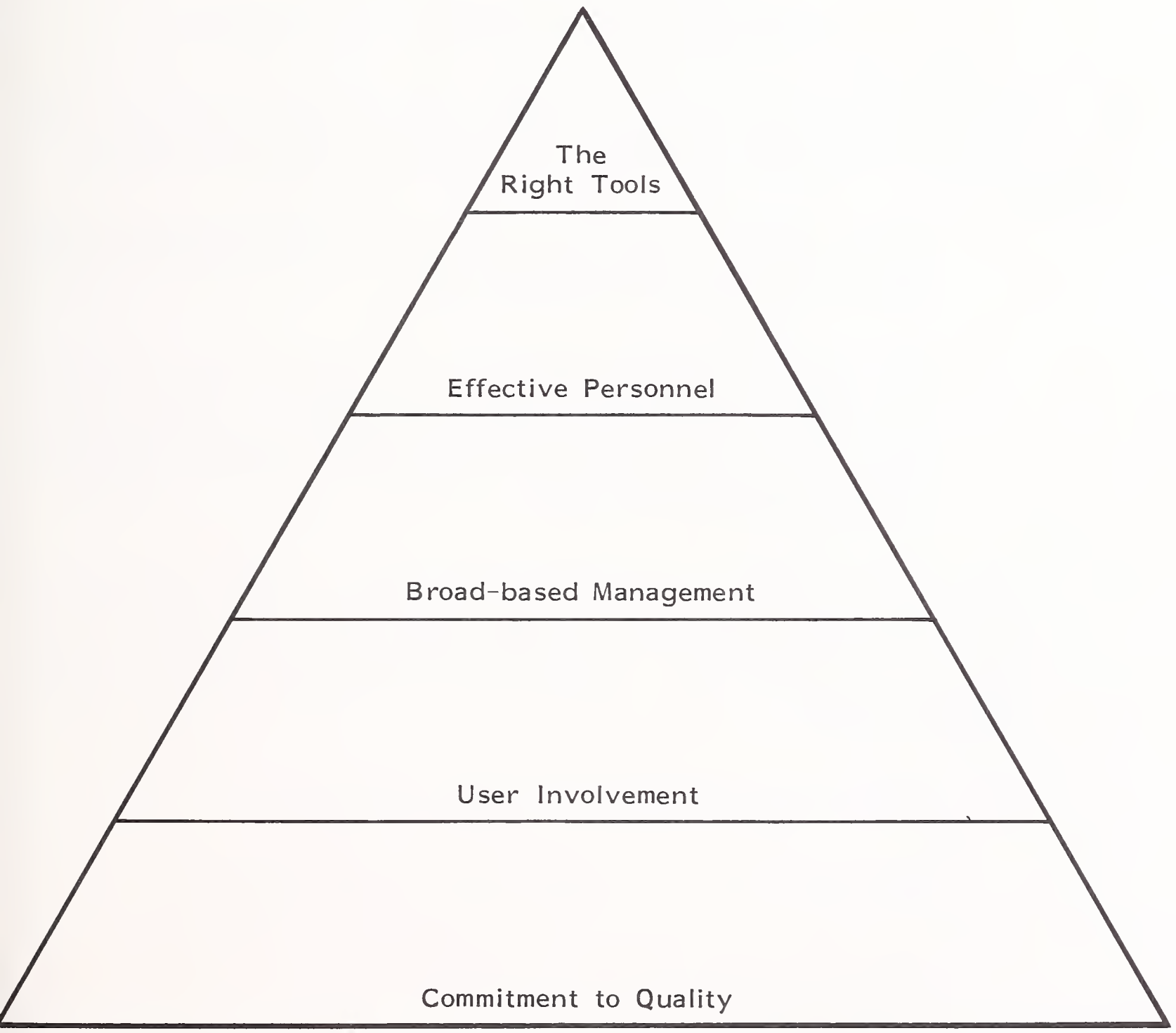
    - Multiple iterations in the prototype phase.

    - User involvement (or, sometimes, construction).

    - User-IS communication via the FGL.

EXHIBIT V-8

THE PRODUCTIVITY PYRAMID



- Flexibility is inherent in the FGL approach.

- FGLs can be supportive of data integrity by:

  . Containing an integral DBMS.

  . Enabling experiments with different data structures.

- One danger inherent in FGLs is that FGL users may become carried away with the ease and flexibility of the tool and not manage data and information.

  - AN FGL should be viewed as an important means of supporting data-based analysis and not a replacement for it.

  - FGLs may unintentionally shift the focus of attention from the data to the process, especially when used by under-trained and under-experienced users. IS analysts skilled in data analysis should be assigned to work with users in this situation.

- An associated problem is that FGLs make it so easy to add all possible bells and whistles that everything that might possibly be of use is added to the system, even if rarely used. Problems caused by adding such low priority features include:

  - Adding to system development and maintenance costs.

  - Increasing complexity, and unnecessarily reducing robustness.

  - Increasing operating costs unnecessarily.

- A danger with FGLs is that the resulting system will be undertested.

- In part, this is because FGL-based systems will, in fact, need less time for testing since fewer errors will be found and these errors will be more easily corrected.

- However, the testing program should be the same as a good testing program for third generation systems.

- One of the main objectives of an FGL test should be to identify requirements that have been omitted. In third generation systems there is often not enough time for this function. However, omitted functions are the most serious errors in a system and often the most difficult to repair.

VI IMPACT OF FOURTH GENERATION LANGUAGES

VI IMPACT OF FOURTH GENERATION LANGUAGES

- The strength of the FGL impact depends largely on whether the FGL is employed on secondary (e.g., Information Center) or primary uses (e.g., mainline production systems).

  - Where the FGL is used for secondary purposes, the overall impact on IS will be generally positive, but moderate.

    . An exception will be if users start to learn how to use the FGL.

    . Here, initially, user satisfaction with the FGL in particular and IS in general greatly increased.

    . However, the danger is that unless IS gets on the FGL bandwagon the users will believe they are ahead of IS (and perhaps they may be!). They may then tell IS how to design and implement corporate systems as Exhibit VI-1 illustrates.

  - The impact of the FGL can be much larger where it is used for mainline systems. One of the biggest impacts an FGL can have on IS is before the FGL has even been used, when it is being considered for use. Frankly, some IS personnel are afraid of FGLs. Exhibit VI-2 lists some of these fears.

EXHIBIT VI-1

FGL USER SATISFACTION WITH INFORMATION SYSTEMS
(WHERE INFORMATION SYSTEMS DOES NOT USE FGLs)

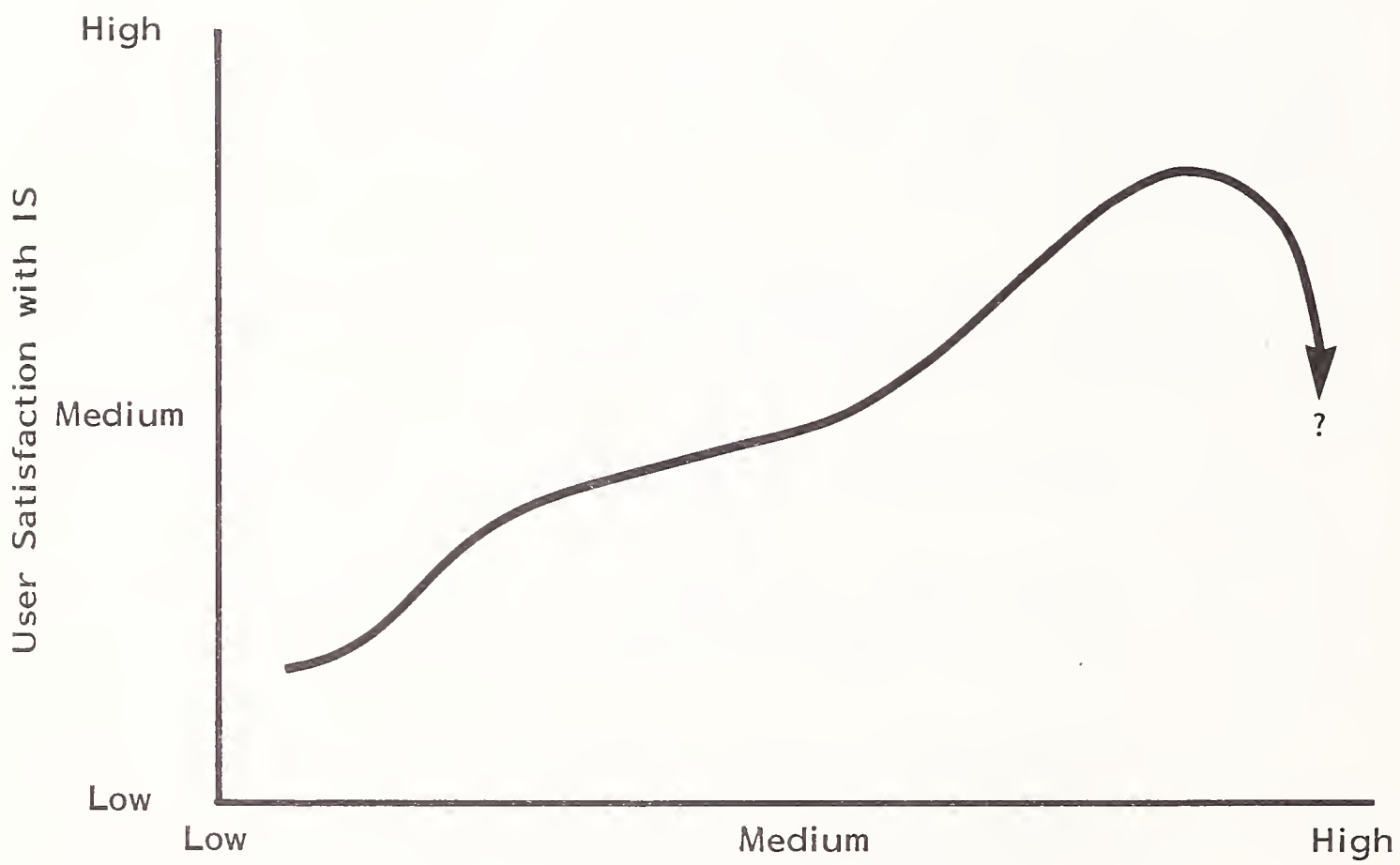


EXHIBIT VI-2

INFORMATION SYSTEMS' FEARS OF FGLs

- No (structured) methodology to replace system development methodology
- No firm divisions between system development steps - difficult to measure progress
- Inefficient use of hardware resources - rapidly rising costs
- Obsoleting of IS technical and management skills
- Programmer opposition to using a FGL
- Loss of power and control by IS
- Users may abandon FGL and leave IS to pick up pieces
- Marginal data processing applications may be implemented using FGLs because of ease/lack of control
- FGLs not technical enough

- These fears relate to real things that can, and usually do, happen after FGL introduction.

- However, these feared problems generally will not be as large as the FGL benefits. It should be recognized, though, that while corporate-wide benefits may be larger than problems this may not be so for individuals (e.g., a COBOL journeyman programmer).

- The remainder of this chapter will:

  - Analyze FGL benefits and their impact.

  - Examine impacts on personnel.

  - Gauge the organizational effects of implementing FGLs.

A. FGL BENEFITS

- IS departments that use FGLs in production systems can reasonably expect these benefits:

  - Faster system implementation.

  - Increased productivity in:

    . Development.

    . Maintenance.

  - Increased system quality.

- Shifting costs from IS to users.

- Backlog reduction, both

  . Visible and

  . Invisible.

- Faster system implementation may only be another aspect of productivity to IS professionals. However, to users this means faster realization of the benefits of a computer system. Often, the first year's projected benefits of a new system are appreciably higher than the system's developmental cost. Sometimes, even more important are the intangible benefits associated with rapid system implementation, e.g., a policy decision, executive promises fulfilled, promotions, etc., made possible by FGL-produced data.

- Increased productivity is a function both of more rapid coding as well as being able to use less experienced people for a particular task.

- Increased quality is a result of prototyping and user communiations. As noted in the last chapter, increased quality is not automatic and requires other actions by IS.

- Shifting costs from IS to users has an obvious impact on IS by having IS' own directly chargeable costs decline. The analogy here is in the transfer of IS data entry (i.e., keypunching) to user areas (i.e., on-line data entry) in the 1970s.

  - The major benefit is not the budget transfer.

  - Rather, it is the ability of user departments to manage their own function and to see the direct relationship between system costs and benefits.

- This approach can be viewed as the ultimate chargeback system. The shifted functions can include:

  . Standalone development by users;

  . Joint development with IS;

  . Maintenance.

- Backlog reduction will certainly occur on at least part of the invisible backlog via IC use of FGLs. Larger backlog reductions will come about by using FGLs within the IS department and for mainline systems.

- Exhibit VI-3 maps these potential FGL benefits against FGL capabilities.

  - Prototyping in conjunction with FGL-based production ("preproduction prototyping") turns out, not surprisingly, to be the most important FGL capability.

  - Standalone prototyping, while not without value, is relatively the least important.

  - Both fast coding and ease of learning are very important, since together they can make users more productive.

  - Note that PC use in and of itself is not of critical importance. However, as discussed earlier, FGL PC use may often facilitate FGL use in production settings.

    . It will certainly make FGL use in computer-intensive analytic work more feasible.

FOURTH GENERATION LANGUAGE CAPABILITIES' IMPACT ON POTENTIAL BENEFITS

FGL POTENTIAL BENEFIT	FGL CHARACTERISTICS						
	EASE OF LEARNING	PROTOYPING CAPABILITIES		FAST. CODING	DATA BASE CONSTRUCTION	REPORT GENERATION	PC USE
		Standalone	Preproduction				
Faster System Implementation	Medium	Medium	High	High	Medium	Medium	Medium
Increased Productivity							
• Development	High	Medium	High	High	High	High	High
• Maintenance	High	Low	High	Medium	High	High	Medium
Increased Quality	High	Medium	High	Medium	Medium	Medium	Medium
Shifting Costs from IS to Users	High	Low	High	High	High	High	High
Backlog Reduction							
• Visible	Medium	Low/Medium	High	High	Medium/High	Medium	Medium
• Invisible	High	High	Medium	High	High	High	High
Number of "Highs"	5	1	6	5	4.5	4	3

. In the long run, PC-based FGL use should replace much FGL timesharing (both internal and external).

B. FGL IMPACT ON PERSONNEL

- FGL use will have significant impact on personnel, both IS and non-IS.
  - User skills will increase significantly, since in hours or days user personnel can be writing their own (simple) programs.
    . Some users may, in fact, become highly skilled in an FGL and will be able to produce quite sophisticated programs.
    . This would make it possible to have meaningful transfers from user areas into IS technical positions.
  - IS skills will be greatly rearranged.
    . Currently the core of most IS staffs is the experienced COBOL programmers.
    . This kind of skill will be made obsolete by FGLs.
    . Experience to date indicates that bright people with little IS background can handle FGLs at least as well as experienced IS staff.
    . In addition, roughly half of existing IS staff will resist using FGLs.

- These points raise some profound issues concerning IS staffing in an FGL environment.

  - Exhibit VI-4 shows a schematic of typical IS career progressions (ignoring departures to and arrivals from outside the enterprise).

  - This structure changes drastically in an FGL environment.

    . Journeymen FGL progammer analysts can be created in a much shorter time.

    . There is no real reason to have a strictly IS-generated pool of such entry level people.

    . People can become expert in an FGL faster than in a third generation language.

    . Much of the skilled personnel can reside in user areas.

    . Exhibit VI-5 shows how the post-FGL personnel environment differs from the current environment.

- There is implicit "deskilling" in some IS jobs as a result of this. Presumably, it will be roughly balanced by the increased skill levels in user areas.

  - IS management must have a strategy for recruiting the right kind of people into FGL jobs. This will often require coordinated recruiting efforts with key user areas.

  - Similarly, IS should plan for the decline in requirements for third generation system personnel. The long maintenance "tail" for most systems guarantees that such jobs will not vanish overnight. Because of the disrepute of maintenance in the minds of many system profes-

EXHIBIT VI-4

CURRENT PERSONNEL ADVANCEMENT
(SCHEMATIC)

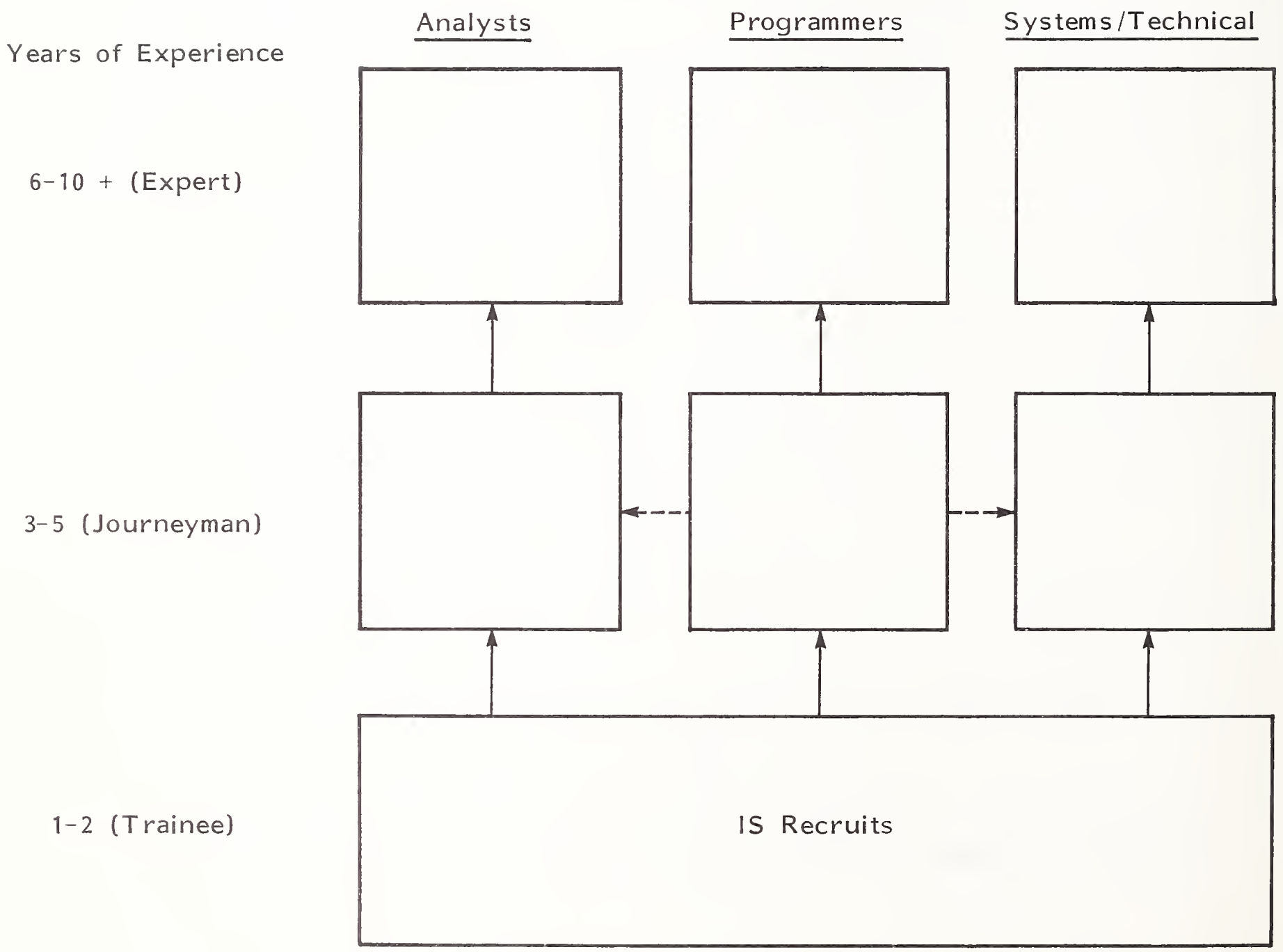
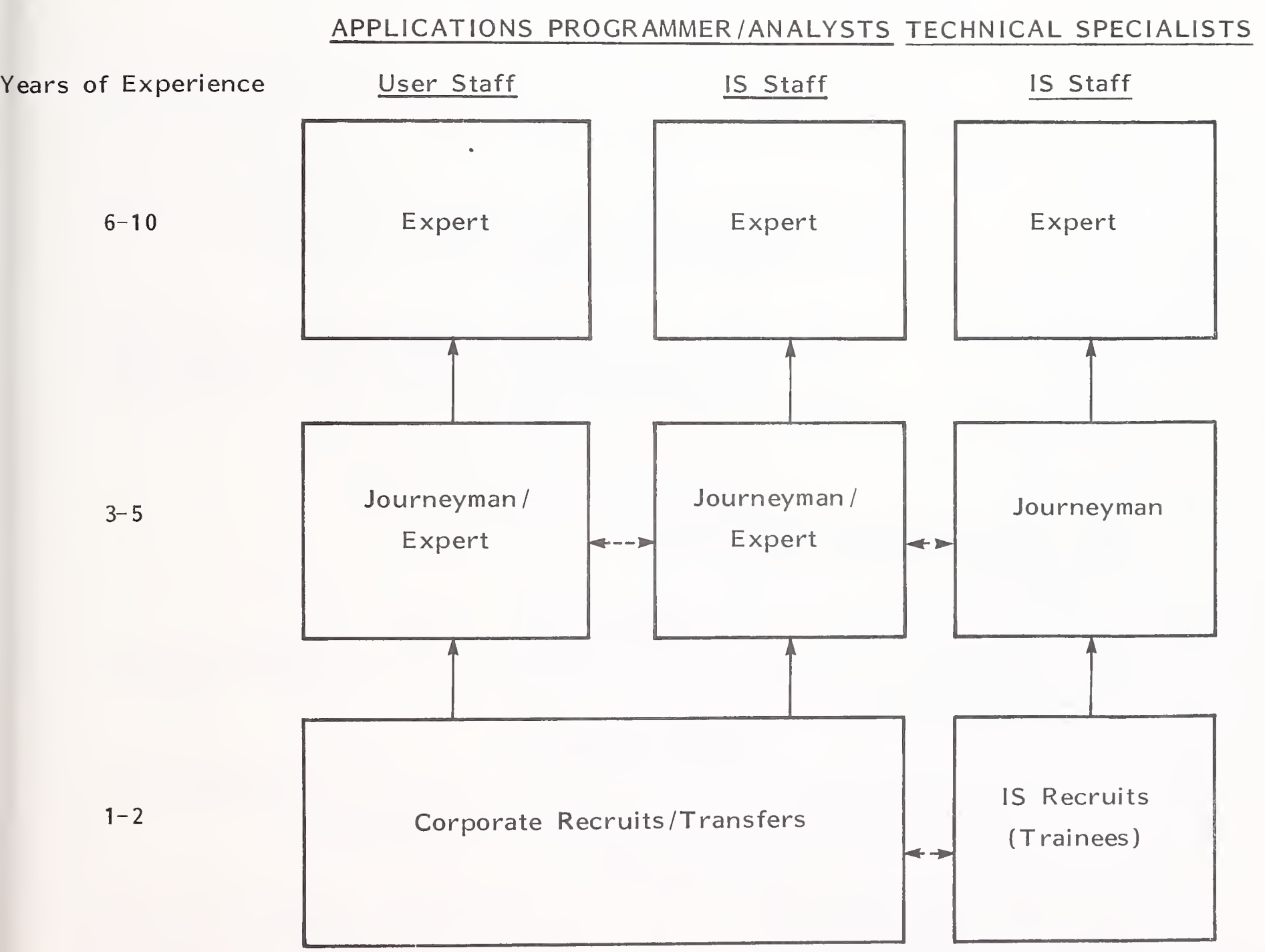


EXHIBIT VI-5

POST-FGL PERSONNEL RELATIONSHIPS



sionals, offering a future in COBOL maintenance will certainly provide an incentive for many programmers to leave voluntarily. (See INPUT's report, Maintenance: The Uninvited Guest, November 1982.)

- On a happier note, the converging of user and IS skills will make it feasible to have significant movement of personnel between users and IS areas.

  - This is a goal that both IS and user management desire, but which is rarely attained, as Exhibits VI-6 and VI-7 illustrate.

  - A routine part of college recruits' training could be a one- or two-year stint in the FGL programming/analysis area.

    . This could provide IS opportunities to find and recruit generalists with whom IS otherwise would not come in contact.

    . More broadly, it would serve to close the widening gap between IS and non-IS departments. (See INPUT's report Evaluating EDP Level of Service, January 1982.)

C. FGL ORGANIZATIONAL IMPACT

- The major FGL impact on the IS organization will be the personnel effects described in the previous section.

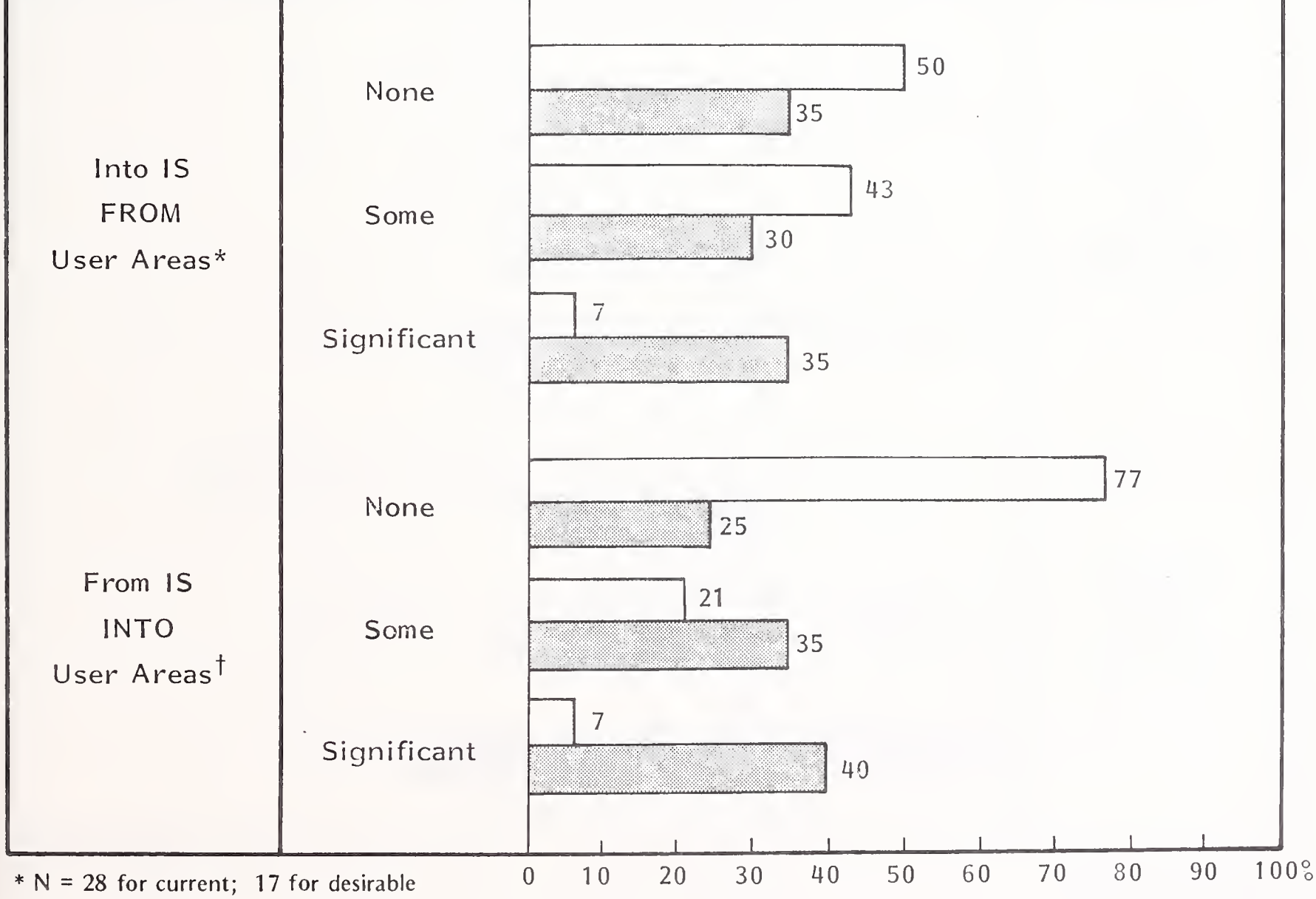
  - Programming and analysis will be much more of a "condominium," with IS and users jointly sharing and administering resources.

  - IS should tend to be the home of the most experienced FGL programmer/analysts. IS will have to make sure that this happens by making sure that it expands its corporate view.

EXHIBIT VI-6

PERSPECTIVE OF INFORMATION SYSTEMS/USER PERSONNEL MOVEMENT: INFORMATION SYSTEMS MANAGEMENT

MOVEMENT	AMOUNT	PERCENT OF RESPONDENTS
Into IS FROM User Areas*	None	50 / 35
	Some	43 / 30
	Significant	7 / 35
From IS INTO User Areas†	None	77 / 25
	Some	21 / 35
	Significant	7 / 40



* N = 28 for current; 17 for desirable

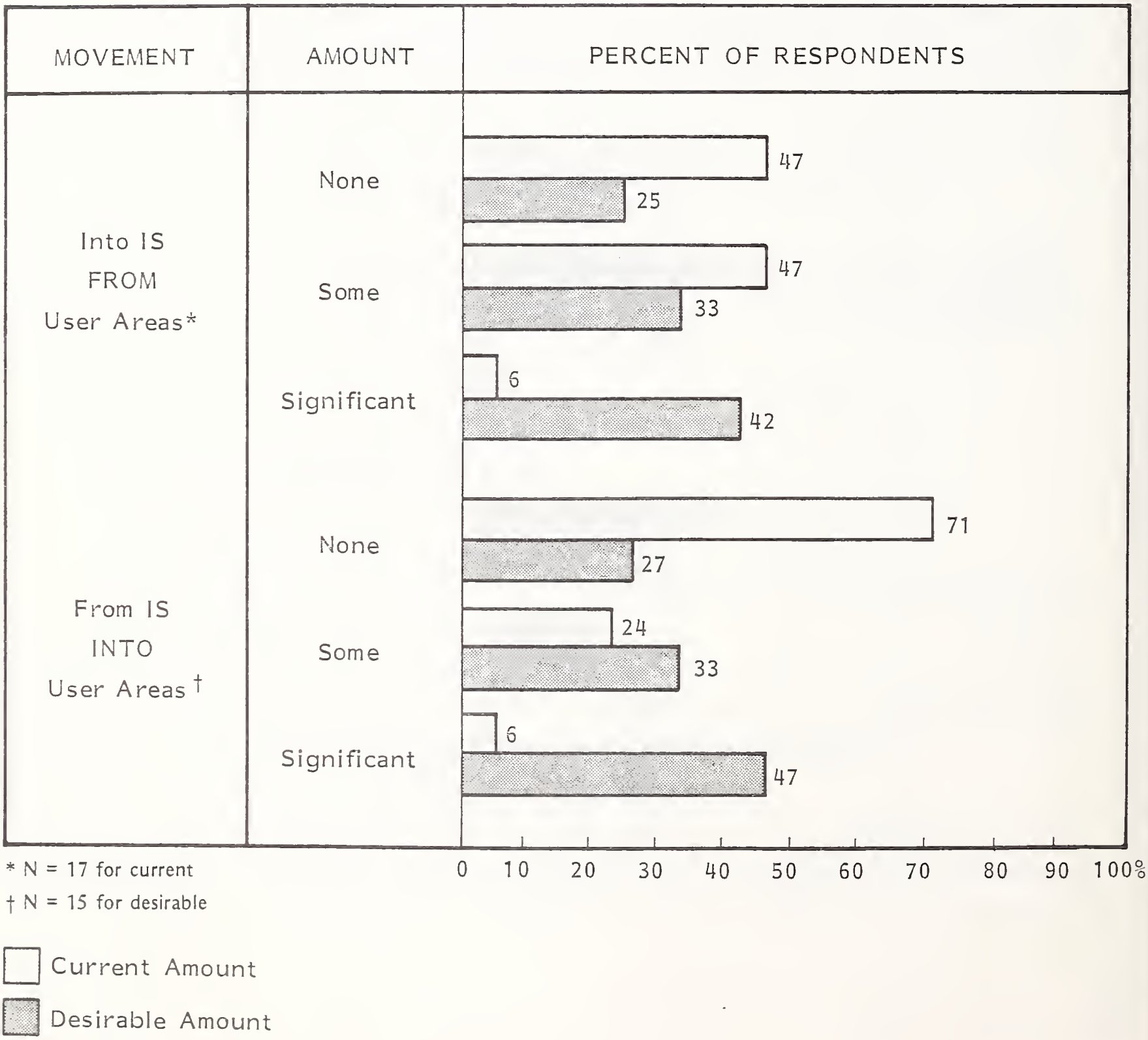
† N = 28 for current; 20 for desirable

Current Amount

Desirable Amount

EXHIBIT VI-7

PERSPECTIVE OF INFORMATION SYSTEMS/USER PERSONNEL MOVEMENT:
KEY USERS



* N = 17 for current

† N = 15 for desirable

Current Amount

Desirable Amount

- For many applications, routine maintenance will be farmed out to user departments.

  - As noted, this will leave IS with residual maintenance responsibilities, largely for third generation systems.

  - IS would provide backstop maintenance as well as a pool of resources to assign to user departments for high priority assignments.

- IS should retain (and usually expand) its quality assurance activities, both for systems it develops as well as those that users develop.

  - The quality assurance group, perhaps in conjunction with the corporation's internal audit group, can draw up guidelines for system development.

  - In addition, the quality assurance group would examine each production system or important analytical program to ensure that it was working as planned.

- In general, IS would become much more of an expert, advisory organization than most IS organizations are now.

  - IS should still retain sufficient system development and operations responsibilities so that it does not forget the real world.

  - However, IS' main role would be to provide planning and technical expertise for the corporation's data processing needs.

VII RECOMMENDATIONS FOR A FOURTH GENERATION LANGUAGE STRATEGY

VII RECOMMENDATIONS FOR A FOURTH GENERATION LANGUAGE STRATEGY

A. STRATEGIC ISSUES

- This report has taken as virtually proven that FGLs have a major role to play in user-driven IC solutions.
  - The real question in INPUT's view is whether FGLs should be taken into the IS department itself and used as a major developmental tool.
  - Since the impact of an FGL, if effective, on IS will be major and lasting, this is not a decision to be made quickly or lightly.
- INPUT believes that each IS organization should examine FGLs very closely as a way to help solve many of its major problems.
  - Used effectively, FGLs can play a major role in alleviating these problems:
    - IS isolation from organizational mainstream.
    - Perceived IS nonresponsiveness to user needs.
    - Obsolete or unsatisfactory systems.

    - User ignorance of data processing.

  - FGLs can also help (but not as much as in the preceding problem area) to resolve:

    - IS organizational complexity.

    - Backlogs.

- Taken together, these problems add up to a crisis for IS in many organizations. FGLs can go a long way to resolve this crisis. Ideally, the result would be as shown in Exhibit VII-1 (which should be contrasted to Exhibit V-3).

- If an IS department is to use an FGL, it will gain many more benefits by going all the way and using the FGL to run production applications. While using an FGL for pro otyping alone has benefits, these are outweighed by having to recode the production version in a third generation language.

  - Many of the speed and productivity benefits of the FGL development are lost.

  - The correct analysis performed by the FGL prototype may be lost or muddied in the transition.

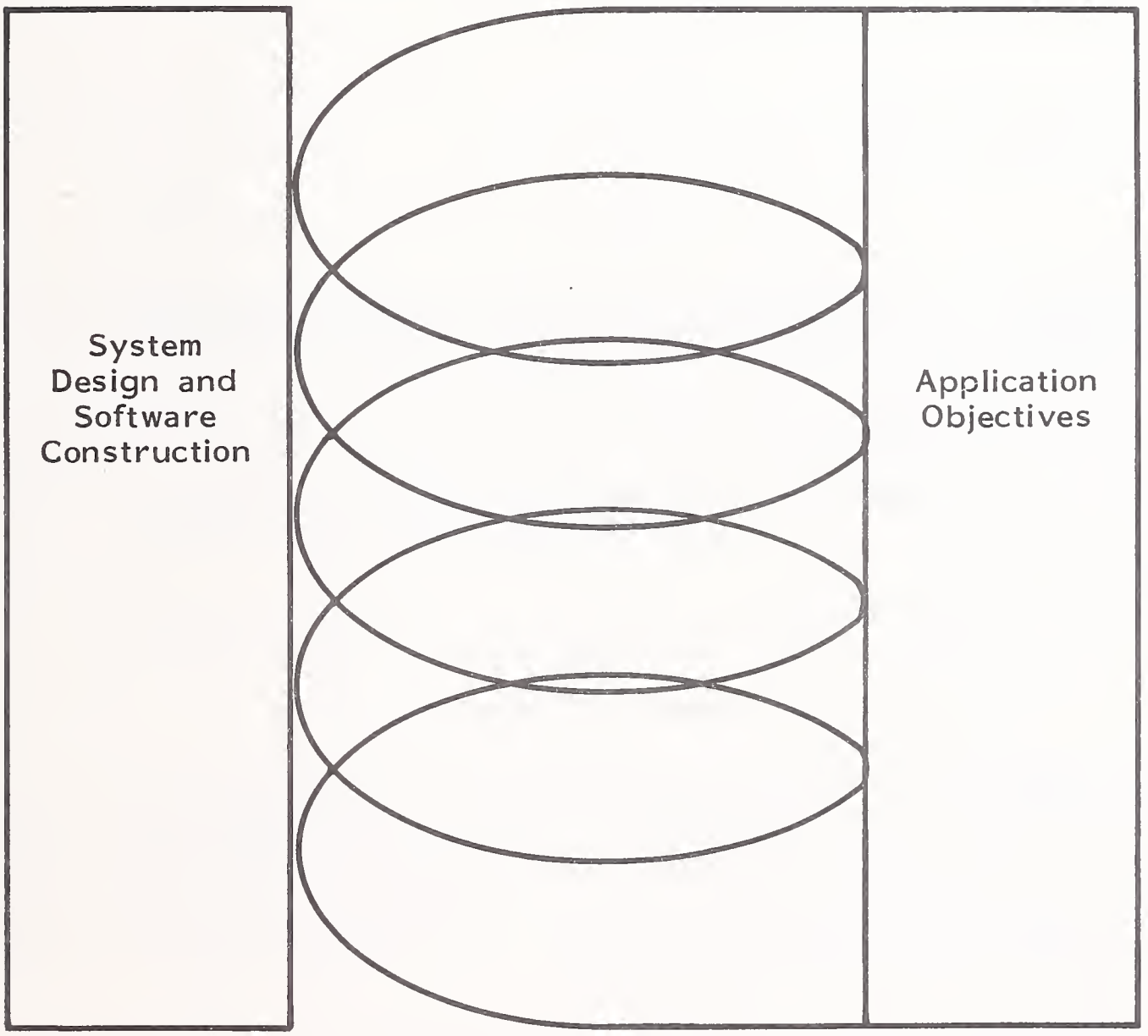
  - The maintenance benefits (of speed and user maintenance) are lost.

- The economics of using an FGL are still open to some question.

  - Taken in isolation, the developmental savings in using an FGL are about the same as increased FGL operating costs.

EXHIBIT VII-1

POSITIVE, ASCENDING FEEDBACK WITH FGL



- However, FGL usage is almost certain to stimulate a significant increase in overall computing use. Some of this increase may be distributed to PCs and super-micros; however, in the medium run (up to three years), IS computing expenses should increase significantly.

- If FGL use lives up to its promise, there should be relatively few organizational complaints about the expense.

B. TACTICAL ISSUES

- IS departments looking to move into FGL environments should use a phased approach, as described in Exhibit VII-2. The most important issues are:

  - Generating accurate cost figures, since specific figures from one's own operation are much more valuable than external figures.

  - Involving users.

  - Thorough, frank briefings for all affected by FGLs.

- The PC implementation of FGLs is going to be among the most exciting events in computing. This should not be a reason to delay introducing FGLs. IS departments that have not already selected an FGL should give great attention to expected PC capabilities.

- The internal effects of FGLs on IS staff will be profound. New career paths will have to be developed for both new and existing staff members.

EXHIBIT VII-2

PHASED FGL IMPLEMENTATION FOR IS PROJECTS

1. Introduce FGL in the Information Center.
2. Build core of IS FGL practitioners.
3. Select a medium-sized (3-5 person year) project where third generation costs (including operating costs) are reasonably certain. A system enhancement is the preferred target.
4. Keep a project diary for future cost and operational analysis.
5. Integrate experience from the first pilot into IS conventional system development methodology.
6. Select a second pilot with the main criteria being the ability to integrate users onto the project team.
7. Develop a company-specific cost model.
8. Assuming the first two pilots are successful, thoroughly brief top management, key user management and IS staff on implications, i.e.:
   - Hardware costs
   - Development approach
   - User rules
   - IS staff rules

APPENDIX A: QUESTIONNAIRE

Questionnaire:
Information Centers and Fourth Generation Languages

1. Information Center Data

A. How long has your company supplied Information Center services? _____ years

a. What was the reason for your company's starting?

_____

_____

b. What type of hardware and software are used?

1. Hardware:

_____

2. Operating System and Communications environment:

_____

3. Software packages made available to users:

_____

_____

_____

B. About how many Information Center users are there now? How many do you expect in two years?

	Number of Users	
Type of User	1983	1985
Programming Staff	_____	_____
Senior Executives	_____	_____
Senior Executives' Staff	_____	_____
Other Managers	_____	_____
Other Managers' Staff	_____	_____
Clerical	_____	_____

2. Please provide estimates on the proportion of your hardware resources that your organization devotes to Information Center activities (now and two years from now). What do you see as the reasons for a change?

	1983	1985	Reason for Change/Comments
• Total Processing capability (express in MIPS*)	_____	_____	
— Percent used for Information Center purposes	_____ %	_____ %	
• Total number of terminals	_____	_____	
— Number of conventional terminals used for Information Center purposes	_____	_____	
• Number shared with other IS functions	_____	_____	
• Number dedicated to Information Center purposes	_____	_____	
• Number of intelligent workstations tied to mainframes*	_____	_____	

*Millions of instructions per second.

3. The Information Center may sometimes be viewed as filling similar needs as standalone personal computers. Please indicate the advantages and disadvantages that you feel the Information Center has compared to standalone personal computers. Please give comments where you:

- See strong advantages or disadvantages.
- Believe your organization's experience may be different than experience generally.
- See changes occurring in the future.

Informations Center's Advantages & Disadvantages Compared to Standalone Personal Computers (check one)

Area of Comparison	Don't Know	Information Center Has: Strong Advantages	Information Center Has: Advantages	Evenly Balanced	Information Center Has: Disadvantages	Information Center Has: Strong Disadvantages	Comments
Costs							
– Initial	___	___	___	___	___	___	
– Ongoing	___	___	___	___	___	___	
Ease of Use	___	___	___	___	___	___	
User Control	___	___	___	___	___	___	
Consistency in Response Time	___	___	___	___	___	___	
Software							
– Availability	___	___	___	___	___	___	
– Flexibility	___	___	___	___	___	___	
Support							
– Needed	___	___	___	___	___	___	
– Available	___	___	___	___	___	___	
Other (describe) _____	___	___	___	___	___	___	

4. The Information Center may sometimes be viewed as filling similar needs as Commercial Timesharing. Please indicate the advantages and disadvantages that you feel the Information Center has compared to Commercial Timesharing. Please give comments where you:

- See strong advantages or disadvantages.
- Believe your organization's experience may be different than experience generally.
- See changes occurring in the future.

Information Center's Advantages & Disadvantages Compared to Commercial Timesharing (check one)

Area of Comparison	Don't Know	Information Center Has: Strong Advantages	Information Center Has: Advantages	Evenly Balanced	Information Center Has: Disadvantages	Information Center Has: Strong Disadvantages	Comments
Costs							
– Initial	___	___	___	___	___	___	
– Ongoing	___	___	___	___	___	___	
Ease of Use	___	___	___	___	___	___	
User Control	___	___	___	___	___	___	
Consistency in Response Time	___	___	___	___	___	___	
Software							
– Availability	___	___	___	___	___	___	
– Flexibility	___	___	___	___	___	___	
Support							
– Needed	___	___	___	___	___	___	
– Available	___	___	___	___	___	___	
Other (describe) _____ _____ _____	___	___	___	___	___	___	

5. Which Fourth Generation Languages do you use or plan to use in your company?

Languages	Year Use Began/ Will Begin	Number of People Using it Now		Number of People Expected to Be Using it in 1985	
		Programmers	Nonprogrammers	Programmers	Nonprogrammers

6. Fourth Generation Language Production and Prototype System

A. What has been your experience in having end users construct Fourth Generation Language based programs that are used for ongoing production or reporting (i.e., replacing conventional systems development)?

B. To what extent are Fourth Generation Languages used to construct prototype systems that are later used to design permanent conventional systems?

C. Do you have future plans to use Fourth Generation Languages as production or prototype systems? Please describe.

7. What are the main departments that use the Information Center and Fourth Generation Languages in your company? What are the main uses? How much do you expect this to change in two years?

	1983	1985
Information Center Users and Uses		
Fourth Generation Language Users and Uses		

8. What impact (positive or negative) have Information Centers and Fourth Generation Languages had (or what will they have) on your company? Please be as specific as possible.

Impact On	Information Center	Fourth Generation Language
EDP Programming Request Backlog		
User Satisfaction		
System Quality		
Programming Time		
Revenues		
Expenses		
Profits		
ROI		
Other		

9. Staff Support

A. How much central staff support is provided now for Information Center and Fourth Generation Languages? In two years?

Central Staff Support	Number of People (in full-time equivalents) 1983	1985
Information Center Support	_____	_____
Fourth Generation Language Support	_____	_____

B. What kinds of backgrounds do/will staff have (e.g., systems programming, application software, MBA) ?

_____

_____

_____

_____

10. What are the most important things you expect to learn at the Conference? (List most important first).

Information Centers	Fourth Generation Languages
1. _____	1. _____
_____	_____
2. _____	2. _____
_____	_____
3. _____	3. _____
_____	_____

11. General Questions

A. How many levels below the CEO is the top information systems executive? _____

a. Is this satisfactory? ☐ Yes ☐ No

b. Why? _____

B. How important is the organization's information systems capability to the CEO?
(DK = Don't Know, 1 = Low Importance, 5 = High Importance) _____

– Why? _____

_____

C. How many levels below the top information systems executive in the top data administrator? _____

– How will this change in the future? _____

_____

D. What is the experience or knowledge of the top data administrator in the following areas? What should it be in the future? (1 = Low, 5 = High)

Experience Area	Amount of Experience Now	Future
General Data Processing	_____	_____
Data Base Management Software	_____	_____
Specific Application(s)	_____	_____
User Department Operations	_____	_____
Other (specify) _____	_____	_____

_____

E. What is used to measure the performance of the information systems function (either generally or for particular parts)?

Measurement	Type of Process (check) Informal	Formal
Internal Rate of Return	☐	☐
Return on Investment	☐	☐
Cost/Benefit	☐	☐
Other (specify) _____	☐	☐

_____

F. Which formal design or programming methodologies do you use or plan to use?

_____

_____

_____

_____

APPENDIX B: ANALYTIC DESCRIPTIONS OF SELECTED FOURTH GENERATION LANGUAGE PACKAGES

APPENDIX B: ANALYTIC DESCRIPTIONS OF SELECTED FOURTH GENERATION LANGUAGE PACKAGES

- Attached are exhibits describing features and plans for the following FGL:

FGL	Exhibit Number
- FOCUS	B-1
- NOMAD 2	B-2
- RAMIS II	B-3
- NATURAL	B-4
- INQUIRE	B-5
- ADS + ON-LINE ENGLISH	B-6
- MARK V	B-7
- SYSTEM W	B-8
- SALVO	B-9

- This information was obtained from vendor publications and interviews. INPUT has endeavored to check its reasonableness and accuracy. However, it cannot be responsible for errors or omissions.

  - Equally important, vendors may change their strategies and product characteristics in this rapidly evolving area.

  - Consequently, prospective users of FGLs should carefully check the accuracy and completeness of FGL product information before coming to a decision.

- Vendor names and addresses are supplied in Exhibit B-1 through B-10.

EXHIBIT B-1

ANALYTIC DESCRIPTION OF FOCUS

GENERAL CHARACTERISTIC	FGL PRODUCT INFORMATION
A. FEATURES	
1. "What-if" capabilities	Yes
2. Spreadsheet capabilities	Yes
3. Built-in financial function capabilities	Additional module
a) Depreciation	
b) Discounted cash flow	
c) Loan amortization	
d) Payback	
4. Built-in statistical functions	Additional Module: Descriptive, crosstabs, correlation regressions, ANOVA, timeseries, factor analysis, discrimination analysis, exponential smoothing
5. Other modeling capabilities	Graphics (color)
a) Monte Carlo (risk analysis)	
b) Goal seeking (backward iteration)	
c) Sensitivity analysis	
d) Simultaneous equations	
B. ADVANTAGES	
1. Productivity Increases	5:1 development days
2. Prototype use	Yes
a) Complex queries	
b) Multiuser update	
3. Use in production systems	Yes-dialogue 4-5 mm record DB's 2-3 minute response e.g., bank customer files, brokerage trading files

Continued

EXHIBIT B-1 (Cont.)

ANALYTIC DESCRIPTION OF FOCUS

GENERAL CHARACTERISTIC	FGL PRODUCT INFORMATION
4. End-user ease of training	
a) Level of understanding	Easy, can be productive first day
b) Time required	2-3 hours to 3 days
c) Amount of training needed	2-3 hours to 3 days
5. Programmer and end-user suitability - Why?	Yes
C. IMPROVEMENTS BEING MADE	
1. Efficiency	DBA can limit number of records retrieved on any request
a) Catalogued processing with parameters	
b) Restructured data base without reload	
c) Secondary index formation	
d) Tuning on the fly	
2. Applications Supported	
a) Integrity	
b) Security	File segment, field, value; encryption of data and procedures
c) Auditability	Up to size log files
3. Interfaces (to what?)	
a) e.g., Stat	
b) Word processors	
c) DD	
d) Graphics	
e) Other 4GL files VSAM, IMS (extract or direct)	

Continued

EXHIBIT B-1 (Cont.)

ANALYTIC DESCRIPTION OF FOCUS

GENERAL CHARACTERISTIC	FGL PRODUCT INFORMATION
4. Other	Relational JOIN of up to 16 VSAM files
a) CICS	Additional module
b) TSO	Additional module
c) VM/CMS	Yes
d) SSX/VSE	No
5. In what ways is software superior?	Totally integrated, offers all functions in one environment
	PC interface
	Others are all parameter-driven, FOCUS can even add screens in the middle of a request
D. SUPPORT FEATURES (Now or end 1984)	
External	
1. Manuals (Titles)	User manual, Query language primer, Course materials
2. Self-study courses (how long)	PC = primer plus user manual
3. CAI/VAI (how much)	CAI uses 115, $275/month or $8,000 purchase
4. Live instruction (student hours/ year)	~35,000 days/year
5. Other	Many other firms offer FOCUS training
Built-in	
6. Optional tutorials branching from "HELPS"	HELP and ERROR facility, no tutorials except CAI
7. Menu versus command (options to choose?)	Either; basic mode is command level, menus must be written
8. Other	

Continued

EXHIBIT B-1 (Cont.)

ANALYTIC DESCRIPTION OF FOCUS

GENERAL CHARACTERISTIC	FGL PRODUCT INFORMATION
E. PC OFFERING	Now
1. Features of mainframe included	All (by stages), Now: Report generator, Dialogue manager
2. Features of mainframe not included	Hard disk limits on file size, 4K schema size, etc.
3. Added to PC, not in mainframe	
4. Compatibility upward/downward	Yes, both within size limitations
5. Code	
6. Files	
7. How does it work (standalone, DDP, other)	Standalone or interactive now, DDP in 1984, not multithread, uses queried synchronous machine to coordinate requests, prevent interlocks
8. Status of development/test/use	Stage 1 now, Stage 2 being tested 3Q83, Stage 3 1Q84
9. Relative benefits versus mainframe	Cost, convenience (4341 port ~$10K/year, PC ~$10K one time)
F. COSTS	
1. Mainframe	Basic system plus CRT data entry language $71,500 or $1,830/month; other features extra
2. Personal computer	$2,200 plus $750 for add-on board (provides 640K addressable memory)
3. Discounts	Yes, by number of units
4. PCs available	
a) IBM	Yes, now
b) O/S	MS/DOS (TI-prof)
5. Any super-micro/turnkey plans	Strictly software, strictly Intel chip

EXHIBIT B-2

ANALYTIC DESCRIPTION OF NOMAD 2

GENERAL CHARACTERISTIC	FGL PRODUCT INFORMATION
A. FEATURES	
1. "What-if" capabilities	Yes: specific verb plus temporary update
2. Spreadsheet capabilities	Yes, full screen editor and/or commands
3. Built-in financial functions capabilities	Yes
a) Depreciation	
b) Discounted cash flow	
c) Loan amortization	
d) Payback	
4. Built-in statistical functions	Yes
5. Other modeling capabilities	Graphics
a) Monte Carlo (risk analysis)	Not presently
b) Goal seeking (backward iteration)	Yes
c) Sensitivity analysis	Yes
d) Simultaneous equations	Yes
B. ADVANTAGES	
1. Productivity increases	IMS is 5:1, 50:1 for report writing (Exxon study)
2. Prototype use	Yes
a) Complex queries	Yes
b) Multiuser update	Yes
3. Use in production systems	4 mm record customer file, 80 simultaneous users
4. End-user ease of training	
a) Level of understanding	Easy, productive, first day
b) Time required	First week tops
c) Amount of training needed	Few hours to 2-3 days

Continued

EXHIBIT B-2 (Cont.)

ANALYTIC DESCRIPTION OF NOMAD 2

GENERAL CHARACTERISTIC	FGL PRODUCT INFORMATION
5. Programmer and end-user suitability - Why?	Both - very powerful, integrated package with single command environment
C. IMPROVEMENTS BEING MADE	
1. Efficiency	
a) Catalogued processing with parameters	Yes
b) Restructured database without reload	Yes
c) Secondary index formation	Could
d) Tuning on the fly	Could
2. Applications supported	
a) Integrity	Shadow and queued update
b) Security	
c) Auditability	Vertical and horizontal plus encypher
3. Interfaces (to what?)	
a) e.g., Stat	Included
b) Word processors	Not really (VM text editor)-
c) DD	Has TEXT datatype
d) Graphics	Included
e) Other FGL files VSAM, IMS extract or direct	Can extract IMS, direct VSAM and QSAM
4. Other	
a) CICS	
b) TSO	MVS/TSO 1Q84
c) VM/CMS	VM now
d) SSX/VSE	

Continued

EXHIBIT B-2 (Cont.)

ANALYTIC DESCRIPTION OF NOMAD 2

GENERAL CHARACTERISTIC	FGL PRODUCT INFORMATION
5. In what ways is software superior?	All commands in single environment Power of data definition language removes much of the worry and effort regarding integrity, reporting formats, etc. Richness of facilities (financial plus DBMS/application)
D. SUPPORT FEATURES (Now or End 1984)	
External	
1. Manuals (Titles)	Full range, McCracken's book
2. Self-study courses (how long)	Customer-developed, may be released
3. CAI/VAI (how much)	ASI, Deltak plus own (Future)
4. Live instruction (student-hours/year)	~ 20,000 days/year (15,000 users in 5 years)
5. Other	
Built-in	
6. Optional tutorials branching from "HELPS"	Being tested now
7. Menu versus command (option to choose?)	Yes, can change within session
E. PC OFFERING	Future announcement
1. Features of mainframe included	
2. Features of mainframe not included	
3. Added to PC not in mainframe	Data plus intelligence at both ends, i.e. error recovery plus data model transformation into native mode (both ways) between, e.g., 1-2-3, dBase II, Knowledgeman, NPL
4. Compatibility upward/downward	Within limitations

Continued

EXHIBIT B-2 (Cont.)

ANALYTIC DESCRIPTION OF NOMAD 2

GENERAL CHARACTERISTIC	FGL PRODUCT INFORMATION
5. Code	
6. Files	See above
7. How does it work (standalone, DDP, other)	Integrated/interactive
8. Status of development/test/use	Being tested
9. Relative benefits versus mainframe	
F. COSTS	
1. Mainframe	$125K, $135K with/FINAL and training
2. Personal Computer	
3. Discounts	
4. PCs available	
a) IBM	Not yet available
b) O/S	
5. Any super-micro/turnkey plans	No

EXHIBIT B-3

ANALYTIC DESCRIPTION OF RAMIS II

GENERAL CHARACTERISTIC	FGL PRODUCT INFORMATION
A. FEATURES	
1. "What-if" capabilities	
2. Spreadsheet capabilities	
3. Built-in financial function capabilities	Yes
a) Depreciation	
b) Discounted cash flow	
c) Loan amortization	
d) Payback	
4. Built-in statistical functions	Simple thins and interface to SAS
5. Other modeling capabilities	Graphics
a) Monte Carlo (risk analysis)	
b) Goal seeking (backward iteration)	
c) Sensitivity analysis	
d) Simultaneous equations	
B. ADVANTAGES	
1. Productivity increases	5:1 development days "30% better than other FGLs", e.g., RAMIS 6 pp:18 lines Cobol code versus RAMIS II code
2. Prototype use	Yes
a) Complex queries	
b) Multiuser update	
3. Use in production systems	1 mm record data base in market research
4. End-user ease of learning	
a) Level of understanding	Easy, productive first day
b) Time required	One week tops
c) Amount of training needed	Few hours to 2½ days

Continued

EXHIBIT B-3 (Cont.)

ANALYTIC DESCRIPTION OF RAMIS II

GENERAL CHARACTERISTIC	FGL PRODUCT INFORMATION
5. Programmer and end-user suitability - Why?	Yes
C. IMPROVEMENTS BEING MADE	
1. Efficiency	
a) Catalogued processors with parameters	
b) Restructure data base without reload	
c) Secondary index formation	
d) Tuning on the fly	
2. Applications support	Not in application business
a) Integrity	
b) Security	
c) Auditability	
3. Interfaces (to what?)	SAS
a) e.g., Stat	
b) Word processors	
c) DD	
d) Graphics	
e) Other FGL Files VSAM, IMS (extract or direct)	
4. Other	
a) CICS	
b) TSO	
c) VM/CMS	
d) SSX/VSE	

Continued

EXHIBIT B-3 (Cont.)

ANALYTIC DESCRIPTION OF RAMIS II

GENERAL CHARACTERISTIC	FGL PRODUCT INFORMATION
5. In what ways is software superior?	Entirely new product from original RAMIS ENGLISH does not require special load of application dictionary (only implementation like that), plus can intermix RAMIS II and ENGLISH commands, don't need to stay in ENGLISH once you get there
D. SUPPORT FEATURES (Now or End 1984)	
External	
1. Manuals (Titles)	User manual
2. Self-study courses (how long)	
3. CAI/VAI (how much)	3Q83 annual lease
4. Live instruction (student-hours/year)	D/K
5. Other	
Built-in	
6. Optional tutorials branching from "HELPS"	HELP and ERROR facility, no tutorials except CAI
7. Menu versus command (option to choose?)	
8. Other	
E. PC OFFERING	Will be out late winter 1984
1. Features of mainframe included	
2. Features of mainframe not included	
3. Added to PC, not in mainframe	
4. Compatibility upward/downward	

Continued

EXHIBIT B-3 (Cont.)

ANALYTIC DESCRIPTION OF RAMIS II

GENERAL CHARACTERISTIC	FGL PRODUCT INFORMATION
5. Code	
6. Files	
7. How does it work (standalone, DDP, other)	
8. Status of development/test/use	
9. Relative benefits versus mainframe	Competitive requirement
F. COSTS	
1. Mainframe	Depends on system size, from $40,000 to $80,000 for basic system, other features extra, monthly lease available
2. Personal computer	Not yet available
3. Discounts	
4. PCs available	
a) IBM	Will be 68K based
b) O/S	4Q84
5. Any super-micro/turnkey plans	

EXHIBIT B-4

ANALYTIC DESCRIPTION OF ADABAS/NATURAL

GENERAL CHARACTERISTIC	FGL PRODUCT INFORMATION
A. FEATURES	
1. "What-if" capabilities	Not at present time
2. Spreadsheet capabilities	Not at present time
3. Built-in financial function capabilities	No
a) Depreciation	
b) Discounted cash flow	
c) Loan amortization	
d) Payback	
4. Built-in statistical functions	Third party interface to SAS
5. Other modeling capabilities	Graphics (requires GDDM)
a) Monte Carlo (risk analysis)	
b) Goal seeking (backward iteration)	
c) Sensitivity analysis	
d) Simultaneous equations	
B. ADVANTAGES	
1. Productivity increases	90% less coding than COBOL, 60% improvement in development time No operational penalty compared to COBOL
2. Prototype use	
a) Complex queries	Yes
b) Multiuser update	Yes
3. Use in production systems	10mm plus DB calls/day, brokerage
4. End-user ease of learning	
a) Level of understanding	
b) Time required	Few hours to two days
c) Amount of training needed	

Continued

EXHIBIT B-4 (Cont.)

ANALYTIC DESCRIPTION OF ADABAS/NATURAL

GENERAL CHARACTERISTIC	FGL PRODUCT INFORMATION
5. Programmer and end-user suitability - Why?	Programmers can develop total applications, End-user facilities can be as simple and restrictive as desired; built-in security and integrity features
C. IMPROVEMENTS BEING MADE	
1. Efficiency	Check on input
a) Catalogued processing with parameters	Yes
b) Restructure data base without reload	Yes
c) Secondary index formation	Yes , O/L
d) Tuning on the fly	
2. Applications supported	
a) Integrity	Auto backout
b) Security	Applicable, function, library, file, field, value
c) Auditability	
3. Interfaces (to what?)	Graphics, SAS
a) e.g., Stat	
b) Word processors	V/D and EM resource scheduling
c) DD	Data manager
d) Graphics	
e) Other FGL files, VSAM, IMS (extract or direct)	VSAM, announced
4. Other	
a) CICS	Yes
b) TSO	Yes
c) VM/CMS	Yes
d) SSX/VSE	Yes, own, some others

Continued

EXHIBIT B-4 (Cont.)

ANALYTIC DESCRIPTION OF ADABAS/NATURAL

GENERAL CHARACTERISTIC	FGL PRODUCT INFORMATION
5. In what ways is software superior?	Ease of use, flexibility, now recognized as significant for relational views High performance, don't need separate production and inquiry systems as IBM, Cullinet do Natural offers single syntax for all applications from high volume production to single simple ad hoc inquiry
D. SUPPORT FEATURES (Now or End 1984)	
External	
1. Manuals (titles)	User Manual
2. Self-study courses (how long)	
3. CAI/VAI (how much)	Future
4. Live instruction (student-hours/year)	D/K
5. Other	
Built-in	
6. Optional tutorials branching from "HELPS"	HELP and ERROR facility is hierarchical
7. Menu versus Command (option to choose?)	Can develop menus, QBE with Super-Natural
8. Other	Catalogued applications
E. PC OFFERING	May announce by end of 1983, deliver in 1984
1. Features of mainframe include	Subset, same syntax
2. Features of mainframe not included	
3. Added to PC, not in mainframe	
4. Compatibility upward/downward	

Continued

EXHIBIT B-4 (Cont.)

ANALYTIC DESCRIPTION OF ADABAS/NATURAL

GENERAL CHARACTERISTIC	FGL PRODUCT INFORMATION
5. Code	
6. Files	
7. How does it work (standalone, DDP, other)	Download data extract, pass query up
8. Status of development/test/use	
9. Relative benefits versus mainframe	Especially suited to OA applications
F. COSTS	
1. Mainframe	$106,000-172,000, Adabus, $40,000-60,000, Natural, $15,000 graphics, lease available
2. Personal computer	
3. Discounts	Not yet available
4. PCs available	
a) IBM	Yes
b) O/S	68K version
5. Any super-micro/turnkey plans	Showed touch screen version at User conference

EXHIBIT B-5

ANALYTIC DESCRIPTION OF INQUIRE

GENERAL CHARACTERISTIC	FGL PRODUCT INFORMATION
A. FEATURES	
1. "What-if" capabilities	
2. Spreadsheet capabilities	Not at present time
3. Built-in financial function capabilities	No
a) Depreciation	
b) Discounted cash flow	
c) Loan amortization	
d) Payback	
4. Built-in statistical functions	Interface to SAS
5. Other modeling capabilities	SAS Graphics
a) Monte Carlo (risk analysis)	
b) Goal seeking (backward iteration)	
c) Sensitivity analysis	
d) Simultaneous equations	
B. ADVANTAGES	
1. Productivity increases	No figures quoted, will furnish client references
2. Prototype use	
a) Complex queries	Yes
b) Multiuser update	Yes
3. Use in production systems	1mm record data base
4. End-user ease of learning	
a) Level of understanding	
b) Time required	Few hours to few days
c) Amount of training needed	

Continued

EXHIBIT B-5 (Cont.)

ANALYTIC DESCRIPTION OF INQUIRE

GENERAL CHARACTERISTIC	FGL PRODUCT INFORMATION
5. Programmer and end-user suitability - Why?	Both - ease of use plus easy maintenance
C. IMPROVEMENTS BEING MADE	
1. Efficiency	
a) Catalogued processing with parameters	Yes
b) Restructure data base without reload	Some (Keys, e.g.,)
c) Secondary index formation	Yes
d) Tuning on the fly	Yes
2. Applications supported	e.g., litigation support system
a) Integrity	Auto backout by DBA
b) Security	Data base, field, record, system, value, function
c) Auditability	Option in data management supervisor
3. Interfaces (to what?)	SAS, SAS graphics
a) e.g., Stat	Included
b) Word processors	DD can include non-INQUIRE data
c) DD	
d) Graphics	Procedural language both ways
e) Other FGL, files, VSAM, IMS (extract or direct)	Via procedural language
4. Other	
a) CICS	Yes
b) TSO	Yes
c) VM/CMS	Yes
d) SSX/VSE	Yes

Continued

EXHIBIT B-5 (Cont.)

ANALYTIC DESCRIPTION OF INQUIRE

GENERAL CHARACTERISTIC	FGL PRODUCT INFORMATION
5. In what ways is software superior?	Text processing/management is unique Have DDP capability, no user requirement yet Can handle larger data bases and more complex queries because built-in path optimizer
D. SUPPORT FEATURES (Now or End 1984)	
External	
1. Manuals (titles)	User manual course materials
2. Self-study courses (how long)	
3. CAI/VAI (how much)	Future
4. Live instruction (student-hours/year)	D/K
5. Other	
Built-in	
6. Optional tutorials branching from "HELPS"	HELP facility hierarchical, extensive in CREATE data base mode, developing now in user language
7. Menu versus Command (option to choose?)	Macro facility to do menus
8. Other	
E. PC OFFERING	(Announce in Oct. 1Q 84 delivery)
1. Features of mainframe included	Wholly owned subsidiary to develop for IBM-PC, 70% of mainframe capability, no features excluded a priori
2. Features of mainframe not included	
3. Added to PC, not in mainframe	

Continued

EXHIBIT B-5 (Cont.)

ANALYTIC DESCRIPTION OF INQUIRE

GENERAL CHARACTERISTIC	FGL PRODUCT INFORMATION
4. Compatibility upward/downward	Probably both ways
5. Code	
6. Files	
7. How does it work (standalone, DDP, other)	Download data extract, pass query up
8. Status of development/test/use	
9. Relative benefits versus mainframe	Competitive requirement
F. COSTS	
1. Mainframe	$50-190K (multisite, multi-CPU, all components)
2. Personal computers	Not yet available
3. Discounts	
4. PCs available	
a) IBM	Yes
b) O/S	
5. Any super-micro/turnkey plans	No

EXHIBIT B-6

ANALYTIC DESCRIPTION OF ADS AND OLE

GENERAL CHARACTERISTIC	FGL PRODUCT INFORMATION
A. FEATURES	
1. "What-if" capabilities	Some with OLE
2. Spreadsheet capabilities	Not on mainframe
3. Built-in financial function capabilities	No
a) Depreciation	
b) Discounted cash flow	
c) Loan amortization	
d) Payback	
4. Built-in statistical functions	OLE: average, cover, bar charts, ranking
5. Other modeling capabilities	
a) Monte Carlo (risk analysis)	
b) Goal seeking (backward iteration)	
c) Sensitivity analysis	
d) Simultaneous equations	
B. ADVANTAGES	
1. Productivity increases	10:1 development time over COBOL
2. Prototype use	
a) Complex queries	
b) Multiuser update	
3. Use in production systems	Yes, major intent
4. End-user ease of learning	ADS: 1-4 days (programmers) OLE: immediately
a) Level of understanding	
b) Time required	
c) Amount of training needed	

Continued

EXHIBIT B-6 (Cont.)

ANALYTIC DESCRIPTION OF ADS AND OLE

GENERAL CHARACTERISTIC	FGL PRODUCT INFORMATION
5. Programmer and end-user suitability - Why?	ADS - No OLE - Yes
C. IMPROVEMENTS BEING MADE	
1. Efficiency	
a) Catalogued processing with parameters	Improved storage management almost 100% compiled code. Equal to COBOL now
b) Restructure data base without reload	Batch now, O/L next release
c) Secondary index formation	Yes
d) Tuning on the fly	Yes
2. Applications supported	Many (99% in ADS)
a) Integrity	Log at record level before and after images, Prefix only if applicable
b) Security	Yes, multilevel (Definition time and run time)
c) Auditability	Yes
3. Interfaces (to what?)	
a) e.g., Stat	No
b) Word processors	Own
c) DD	
d) Graphics	Graphics via transporter
e) Other FGL, files, VSAM, IMS, (extract or direct)	VSAM
4. Other	
a) CICS	Yes
b) TSO	Yes
c) VM/CMS	Next release
d) SSX/VSE	

Continued

EXHIBIT B-6 (Cont.)

ANALYTIC DESCRIPTION OF ADS AND OLE

GENERAL CHARACTERISTIC	FGL PRODUCT INFORMATION
5. In what ways is software superior?	Easier to use, can define and implement top down through application control facility, i.e., define business function, prototype, then add additional dialogues top down. Sophisticated security features at both definition time and run time.
D. SUPPORT FEATURES (Now or End 1984)	
External	
1. Manuals (titles)	User manual course materials
2. Self-study courses (how long)	
3. CAI/VAI (how much)	Videos (own)
4. Live instruction (student hours/year)	D/K
5. Other	
Built-in	
6. Optional tutorials branching from "HELPS"	Now - available responses only. Tutorials in future release
7. Menu versus Command (option to choose?)	OLE - prompted command
8. Other	
E. PC OFFERING	New product, Knowledgeman, due out by end of 1983
1. Features of mainframe include	Will extend features considerably
2. Features of mainframe not included	
3. Added to PC, not in mainframe	Text processing, spreadsheet, modeling, graphics, can move data back and forth between mainframe

Continued

EXHIBIT B-6 (Cont.)

ANALYTIC DESCRIPTION OF ADS AND OLE

GENERAL CHARACTERISTIC	FGL PRODUCT INFORMATION
4. Compatibility upward/downward	
5. Code	No
6. Files	Yes
7. How does it work (standalone, DDP, other)	Integrated
8. Status of development/test/use	Being tested, due out end of 1983
9. Relative benefits versus mainframe	Greatly extended capabilities
F. COSTS	
1. Mainframe	ADS $40K (plus IDMS $82K*† IDD $35K) OLE $55K
2. Personal computer	Requires IDB $75K plus $1K/copy for PC version
3. Discounts	
4. PCs available	
a) IBM	Yes
b) O/S	MS-DOS 2.0
5. Any super-micro/turnkey plans	

* Includes central code
† in addition

EXHIBIT B-7

ANALYTIC DESCRIPTION OF MARK V

GENERAL CHARACTERISTIC	FGL PRODUCT INFORMATION
A. FEATURES	
1. "What-if capabilities	No
2. Spreadsheet capabilities	No
3. Built-in financial function capabilities	No
a) Depreciation	
b) Discounted cash flow	
c) Loan amortization	
d) Payback	
4. Built-in statistical functions	No
5. Other modeling capabilities	No
a) Monte Carlo (risk analysis)	
b) Goal seeking (backward iteration)	
c) Sensitivity analysis	
d) Simultaneous equations	
B. ADVANTAGES	
1. Productivity increases	5:1 or better development time, no operational penalty (may be improvement) over COBOL
2. Prototype use	
a) Complex queries	Yes
b) Multiuser update	Yes, via IMS
3. Use in production systems	Yes, major intent
4. End-user ease of learning	
a) Level of understanding	If somewhat technical
b) Time required	4-5 day course plus experience
c) Amount of training needed	(need to understand concept)

Continued

EXHIBIT B-7 (Cont.)

ANALYTIC DESCRIPTION OF MARK V

GENERAL CHARACTERISTIC	FGL PRODUCT INFORMATION
5. Programmer and end-user suitability - Why?	Primarily programmer, and/or analyst
C. IMPROVEMENTS BEING MADE	
1. Efficiency	
a) Catalogued processing with parameters	Yes
b) Restructure data base without reload	N/A
c) Secondary index formation	N/A
d) Tuning on the fly	
2. Applications supported	Not presently
a) Integrity	Same as IMS
b) Security	Same as IMS
c) Auditability	Same as IMS
3. Interfaces (to what?)	None directly
a) e.g., Stat	
b) Word processors	
c) DD	
d) Graphics	
e) Other FGL, files, VSAM, IMS (extract or direct)	VSAM 4Q83
4. Other	
a) CICS	4Q83
b) TSO	Yes
c) VM/CMS	No
d) SSX/VSE	Will be

Continued

EXHIBIT B-7 (Cont.)

ANALYTIC DESCRIPTION OF MARK V

GENERAL CHARACTERISTIC	FGL PRODUCT INFORMATION
5. In what ways is software superior?	Compiler performance Can access any files, not only native DBMS Builds own logical views based on user-defined screens Portability to non-IMS environment via "checking a box"
D. SUPPORT FEATURES (Now or End 1984)	
External	
1. Manuals (titles)	User manual course materials
2. Self-study courses (how long)	No
3. CAI/VAI (how much)	Not at present
4. Live instruction (student hours/year)	D/K
5. Other	
Built-in	
6. Optional tutorials branching from "HELPS"	
7. Menu versus Command (option to choose?)	
8. Other	
E. PC OFFERING	Not at present but compare
1. Features of mainframe include	ANSWER/DB plus VisiCalc for related product
2. Features of mainframe not included	
3. Added to PC, not in mainframe	
4. Compatibility upward/downward	

Continued

EXHIBIT B-7 (Cont.)

ANALYTIC DESCRIPTIONS OF MARK V

GENERAL CHARACTERISTIC	FGL PRODUCT INFORMATION
5. Code	
6. Files	
7. How does it work (standalone, DDP, other)	
8. Status of development/test/use	
9. Relative benefits versus mainframe	
F. COSTS	
1. Mainframe	About $100K CICS version will be less
2. Personal computer	N/A
3. Disounts	Yes
4. PCs available	
a) IBM	N/A
b) O/S	
5. Any super-micro/turnkey plans	No

EXHIBIT B-8

ANALYTIC DESCRIPTION OF SYSTEM W

GENERAL CHARACTERISTIC	FGL PRODUCT INFORMATION
A. FEATURES	
1. "What-if" capabilities	Extensive
2. Spreadsheet capabilities	Yes, command or full-screen
3. Built-in financial function capabilities	Yes
a) Depreciation	
b) Discounted cash flow	
c) Loan amortization	
d) Payback	
4. Built-in statistical functions	Simple statistics plus curve fitting, multiple and stepwise regression now; 4Q83 will add correlation, auto correlation, moving average
5. Other modeling capabilities	Graphics (requires GDDM)
a) Monte Carlo (risk analysis)	No
b) Goal seeking (backward iteration	Yes
c) Sensitivity analysis	Yes
d) Simultaneous equations	Yes
B. ADVANTAGES	
1. Productivity increases	No figures quoted, will furnish client with references
2. Prototype use	
a) Complex queries	Yes
b) Multiuser update	Yes, via VM/CMS
3. Use in production systems	Retrieval/modeling system

Continued

EXHIBIT B-8 (Cont.)

ANALYTIC DESCRIPTION OF SYSTEM W

GENERAL CHARACTERISTIC	FGL PRODUCT INFORMATION
4. End-user ease of learning	Easy, few hours to few days if understand modeling 2-day micro-based course
a) Level of understanding	
b) Time required	
c) Amount of training needed	
5. Programmer and end-user suitability - Why?	Intended for end users; programmers not required
C. IMPROVEMENTS BEING MADE	
1. Efficiency	
a) Catalogued processing with parameters	Yes
b) Restructure data base without reload	4-9 viewpoints
c) Secondary index formation	N/A
d) Tuning on the fly	N/A
2. Applications supported	
a) Integrity	Uses MVS/TSO or VM/CMS facilities
b) Security	Uses MVS/TSO or VM/CMS facilities
c) Auditability	N/A
3. Interfaces (to what?)	
a) e.g., Stat	(Timesharing)
b) Word processors	No
c) DD	GDDM HBM
d) Graphics	or EGOS (timesharing)
e) Other FGL, files, VSAM, IMS (extract or direct)	Any extract

Continued

EXHIBIT B-8 (Cont.)

ANALYTIC DESCRIPTION OF SYSTEM W

GENERAL CHARACTERISTIC	FGL PRODUCT INFORMATION
4. Other	
a) CICS	No
b) TSO	Yes
c) VM/CMS	Yes
d) SSX/VSE	No
5. In what ways is software superior?	English-based rules make system easier to learn
	Intelligence of system performs many procedures (e.g., sequence of consolidation) automatically
	First distributed micro-based system
D. SUPPORT FEATURES (Now or End 1984)	
External	
1. Manuals (titles)	User manual
2. Self-study courses (how long)	
3. CAI/VAI (how much)	Micro-based learning station (Apple, IBM-PC) $4,500 hdw, $2,590 courseware
4. Live instruction (student hours/year)	D/K
Built-in	
6. Optional tutorials branching from "HELPS"	Yes
7. Menu versus Command (option to choose?)	Abbreviated commands and menu-driven
8. Other	
E. PC OFFERING	Now
1. Features of mainframe included	2-D only, mainframe is multidimensional

Continued

EXHIBIT B-8 (Cont.)

ANALYTIC DESCRIPTION OF SYSTEM W

GENERAL CHARACTERISTIC	FGL PRODUCT INFORMATION
2. Features of mainframe not included	Otherwise totally compatible
3. Added to PC, not in mainframe	
4. Compatibility upward/downward	
5. Code	Yes, both within 2-D limits. Can download more complex models, than edit locally
6. Files	
7. How does it work (standalone, DDP, other)	Distributed, requires mainframe product
8. Status of development/test/use	Available since Jan. 1983
9. Relative benefits versus mainframe	Convenience; can access library of routines, models, and data
F. COSTS	
1. Mainframe	$55K basic version (4 viewpoints) to $80K full version (9 viewpoints) including 2 micro copies
2. Personal computer	$995 per additional micro version
3. Discounts	On micro version up to 50%
4. PCs available	
a) IBM	Yes
b) O/S	CP/M-86 or MS-DOS 2.0
5. Any super-micro/turnkey plans	No

EXHIBIT B-9

ANALYTIC DESCRIPTION OF SALVO

GENERAL CHARACTERISTIC	FGL PRODUCT INFORMATION
A. FEATURES	
1. "What-if" capabilities	Yes
2. Spreadsheet capabilities	No
3. Built-in financial function capabilities	
a) Depreciation	Template
b) Discounted cash flow	
c) Loan amortization	
d) Payback	
4. Built-in statistical functions	No
5. Other modeling capabilities	No
a) Monte Carlo (risk analysis)	
b) Goal seeking (backward iteration)	
c) Sensitivity analysis	
d) Simultaneous equations	
B. ADVANTAGES	
1. Productivity increases	100:1 COBOL development time, 4:1 other micro-based DBMS both development and performance plus natural language front end
2. Prototype use	
a) Complex queries	Yes
b) Multiuser update	Maybe; write lockouts possible
3. Use in production systems	Within disk storage limitations
4. End-user ease of learning	
a) Level of understanding	Few hours
b) Time required	
c) Amount of training needed	

Continued

EXHIBIT B-9 (Cont.)

ANALYTIC DESCRIPTION OF SALVO

GENERAL CHARACTERISTIC	FGL PRODUCT INFORMATION
5. Programmer and end-user suitability - Why?	Primarily end user, but also micro software application developer
C. IMPROVEMENTS BEING MADE	
1. Efficiency	
a) Catalogued processing with parameters	Yes
b) Restructure data base without reload	Yes
c) Secondary index formation	Yes
d) Tuning on the fly	-
2. Applications supported	Not presently
a) Integrity	Before and after images stored, operates on session files
b) Security	Field, file by user ID
c) Auditability	Possible
3. Interfaces (to what?)	
a) e.g., Stat	Planned
b) Word processor	
c) DD	
d) Graphics	Planned
e) Other FGL files, VSAM, IMS (extract or direct)	Not directly
4. Other	
a) CICS	CP/M
b) TSO	MS/DOS
d) VM/CMS	(PC/DOS)
d) SSX/VSE	

Continued

EXHIBIT B-9 (Cont.)

ANALYTIC DESCRIPTION OF SALVO

GENERAL CHARACTERISTIC	FGL PRODUCT INFORMATION
5. In what ways is software superior?	Advanced features for micro: Virtual JOIN, natural language interpreter, etc. High performance
D. SUPPORT FEATURES (Now or End 1984)	
External	
1. Manuals (titles)	User manual tutorial templates
2. Self-study courses (how long)	Tutorial
3. CAI/VAI (how much)	No
4. Live instruction (student hours/year)	No
5. Other	
Built-in	
6. Optional tutorials branching from "HELPS"	HELP facility
7. Menu versus Command (option to choose?)	3 levels, can be changed mid-session
8. Other	
E. PC OFFERING	Now
1. Features of mainframe included	Not available on mainframe
2. Features of mainframe not included	N/A
3. Added to PC, not in mainframe	N/A
4. Compatibility upward/downward	Across operating systems
5. Code	
6. Files	
7. How does it work (standalone, DDP, other)	Standalone or extract files

Continued

EXHIBIT B-9 (Cont.)

ANALYTIC DESCRIPTION OF SALVO

GENERAL CHARACTERISTIC	FGL PRODUCT INFORMATION
8. Status of development/test/use	Beta test, release in Sept. 1983
9. Relative benefits versus mainframe	Inexpensive
F. COSTS	
1. Mainframe	N/A
2. Personal computers	About $500
3. Discounts	
4. PCs available	
a) IBM	Yes
b) O/S	CP/M, MS-DOS, others
5. Any super-micro/turnkey plans	No

EXHIBIT B-10

VENDOR NAMES AND ADDRESSES

- ADS and ON-LINE ENGLISH, Cullinet Software, Inc. (formerly Cullinane Database Systems, Inc.) 400 Blue Hill Dr., Westwood, MA 02090.
  (617) 329-7700

- DSS/F, Ferox Microsystems Inc., 1701 N. Ft. Myer Dr., Suite 611, Arlington, VA 22209.
  (703) 841-0800

- EASYTRIEVE, Pansophic Systems Inc., 709 Enterprise Drive, Oak Brook, IL 60521.
  (312) 986-6000

- EXPRESS, Management Decision Systems, Inc., 200 5th Ave., Waltham, MA 02254
  (617) 890-1100

- FOCUS, Information Builders, 1250 Broadway, New York, NY 10001.
  (212) 736-4433.

- IDEAL, Applied Data Research, Route 206 and Orchard Lake Road, Princeton, NJ 08540.
  (201) 874-9000.

- IFPS, Execucom, P.O. Box 9578, Austin, TX 78766.
  (800) 531-5038

- INFO, Henco, Inc., 100 5th Ave., Waltham, MA 02254
  (617) 890-8670

- INQUIRE, Infodata, 5205 Leesburg Pike, Falls Church, VA 22041.
  (703) 578-3430

- NATURAL, Software AG of North America, 11800 Sunrise Valley Drive, Reston, VA 22091.
  (703) 860-5050.

- NOMAD 2, National CSS, 187 Danbury Road, Wilton, CT 06897. (203) 762-2511

EXHIBIT B-10 (Cont.)

VENDOR NAMES AND ADDRESSES

- RAMIS II, Mathematica Products Group, P.O. Box 2392, Princeton, NJ 08540.
  (609) 799-2600.

- SALVO, Software Automation Inc., 14333 Proton Rd. Dallas, TX 75234
  (214) 392-3802

- SQL, QBE, IBM, contact IBM at your local sales office.

- SYSTEM W, Comshare, Inc., 3001 State Str., Ann Arbor, MI 48106
  (313) 994-4800.

- VISICALC, Visicorp Inc., 2985 Zanker Road, San Jose, CA 95134
  (408) 946-9000.

APPENDIX C: QUANTIFYING THE COST EFFECTS OF FOURTH GENERATION LANGUAGE USE IN A PRODUCTION ENVIRONMENT

APPENDIX C: QUANTIFYING THE COST EFFECTS OF FOURTH GENERATION LANGUAGE USE IN A PRODUCTION ENVIRONMENT

- The use of FGLs in a production environment has had the greatest impact on programmer and analyst time as well as machine resources used.

  - There are many reports of increases from five to over ten times in programmer and analyst production. It will be assumed here that a productivity increase of five is a practical objective.

  - Under most circumstances hardware resources will not be utilized as efficiently as when using traditional procedure-oriented software. (Although some FGL vendors claim that their FGL code is no less efficient than COBOL.)

- Hardware efficiency is especially difficult to quantify in pre- and post-FGL terms since there have been no realistic large-scale parallel developments of the same system comparing the development of non-FGLs (e.g., COBOL and IMS) and FGLs.

  - However, based on user and vendor estimates and small-scale comparisons, INPUT has made the conservative assumption that FGLs on the average appear to consume about 75% more computing resources than comparable non-FGL tools.

- Most of this excess is concentrated in the processor and main memory requirements, rather than in DASD storage requirements.

- Exhibit C-1 quantifies these assumptions, using a simplified breakdown of expense categories.

  - It is assumed that IS programming and analyst staff could be reduced by 80. Management and operators' responsibilities would be virtually unchanged, however.

  - Processor-related hardware costs would increase by 75% (rounded up to a 1.8 factor increase).

  - Other hardware costs (primarily DASD) would probably increase a nominal amount.

  - For simplicity, it is assumed that other non-personnel expenses would not increase. It is possible that communications costs could increase somewhat, depending on patterns of network use.

- Expense category proportions can vary significantly due to industry, company size, financing arrangements and computing policies; see INPUT's 1982 Annual Report. Companies should adjust this model to take into account their particular position; however, such adjustments will generally have little effect on the conclusions reached.

- The personnel savings and the increased hardware utilization is nearly a "wash" with, in fact, some savings indicated. This is a "steady-state" model: it assumes that all personnel and machine resources are devoted to FGL work. There would obviously be a phase-in and some people and machines would always be devoted to pre-FGL operation. Due to the fact that total resources expended are about the same in pre- and post-FGL environments, this factor is not significant.

EXHIBIT C-1

FOURTH GENERATION COST IMPACT ON I.S. EXPENSE PATTERNS
(AT SAME LEVELS OF PROCESSED OUTPUT)

EXPENSE CATEGORIES	I.S. EXPENSES: Pre-FGL	FGL Effect	I.S. EXPENSES: Post-FGL
Personnel Expense			
Programmers and Analysts	.30	x0.2	.06
Operations and Support	.15	-	.15
Management and Administration	.05	-	.05
Total Personnel	.50	-	.26
Nonpersonnel expense			
Processor-related Hardware	.15	x1.8	.27
Nonprocessor Hardware	.15	x1.2	.18
Communications-related	.10	-	.10
Facilities and Utilities	.10	-	.10
Total Non-personnel	.50	-	.65
Grand Total	1.00	-	.91

- A critical assumption is that IS would be content to produce the same amount of programming and analysis, i.e., reduce the absolute number of IS programmers and redeploy the staff resources elsewhere. This assumption may often be invalid. However, if the IS programming and analysis staff is kept at the same size and produces output at the increased FGL rate, (i.e., five times as much work would be processed), hardware expense will soon be very high. Management and communications expenses would also increase significantly.

  - At steady state, IS expenses would be on the order of three times as great as in a non-FGL environment, as shown in Exhibit C-2.

  - This increased IS expense could be reduced by encouraging FGL systems to be operated on decentralized equipment funded by user departments.

  - In principle, this could reduce IS hardware expense back to the level of the "post-FGL" amount shown in Exhibit C-1.

- If users are encouraged to work with IS to develop systems, then the system development capacity could double (with minimal cost increases to IS), but hardware resources would also double.

  - If the hardware resources were the mainframe type that are traditionally funded from IS budgets, then overall IS expense could increase by more than a factor of five, compared to pre-FGL levels; this is largely due to the increased amount of processing requirements produced by higher application programmer/analyst output, as Exhibit C-3 illustrates.

EXHIBIT C-2

FOURTH GENERATION COST IMPACT ON I.S. EXPENSE PATTERNS
(AT SAME LEVELS OF I.S. STAFF)

EXPENSE CATEGORIES	I.S. EXPENSES: Pre-FGL	FGL Effect	I.S. EXPENSES: Post-FGL
Personnel Expense			
Programmers and Analysts	.30	-	.30
Operations and Support	.15	-	.15
Management and Administration	.05	x2	.10
Total Personnel	.50	-	.55
Nonpersonnel expense			
Processor-related Hardware	.15	x(18x5)	1.35
Nonprocessor Hardware	.15	x(1.2x5)	.90
Communications-related	.10	x2	.20
Facilities and Utilities	.10	-	.10
Total Non-personnel	.50	-	2.45
Grand Total	1.00	-	3.00

EXHIBIT C-3

FOURTH GENERATION COST IMPACT ON I.S. EXPENSE PATTERNS
(USERS CONTRIBUTE TO PRODUCTION SYSTEMS IMPLEMENTATION)

EXPENSE CATEGORIES	I.S. EXPENSES: Pre-FGL	FGL Effect	I.S. EXPENSES: Post-FGL
Personnel Expense			
Programmers and Analysts	.30	-	.50
Operations and Support	.15	-	.15
Management and Administration	.05	x3	.15
Total Personnel	.50	-	.65
Nonpersonnel expense			
Processor-related Hardware	.15	x(1.8x10)	x2.70
Nonprocessor Hardware	.15	x(1.2x10)	x1.80
Communications-related	.10	x2	.20
Facilities and Utilities	.10	-	.10
Total Non-personnel	.50	-	4.80
Grand Total	1.00	-	5.40