SOFTWARE MAINTENANCE PLANNING

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, communications, and office products and services.

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

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

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

OFFICES

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

Dallas
Campbell Center II
8150 N. Central Expressw
Dallas, Texas 75206
(214) 691-8565

New York
Park 80 Plaza West-I
Saddle Brook, New Jerse
(201) 368-9471

United Kingdom
INPUT, Ltd.
Airwork House (4th Floo
35 Piccadilly
London, W.1.
England
01-439-4442
Telex 269776

AFFILIATES

Australia
Infocom Australia

Sweden
P.O. Persson Konsult AB
Box 221 14
Hantverkargatan 7
104 22 Stockholm
Sweden
08-52 07 20

INPUT Planning Services for Management

SOFTWARE MAINTENANCE PLANNING

AUGUST 1982

SOFTWARE MAINTENANCE PLANNING

CONTENTS

Page

SOFTWARE MAINTENANCE PLANNING

EXHIBITS

Page

I INTRODUCTION

I INTRODUCTION

A. BACKGROUND

- The software product and support market is the most rapidly growing sector within the very dynamic computer services market, as shown in Exhibit I-1.

  - Software products were less than 10% of the total computer services market in 1970.

  - By 1986, software products should make up over 28% of the market.

- The markets for both applications and systems software will be growing rapidly, as shown in Exhibit I-2. Systems software will be growing the most rapidly.

  - Systems software includes not only compilers operating systems, but such support software as data base management systems (DBMS), operations support products, report generators, and communications monitors (for a full definition, see Section D of this chapter).

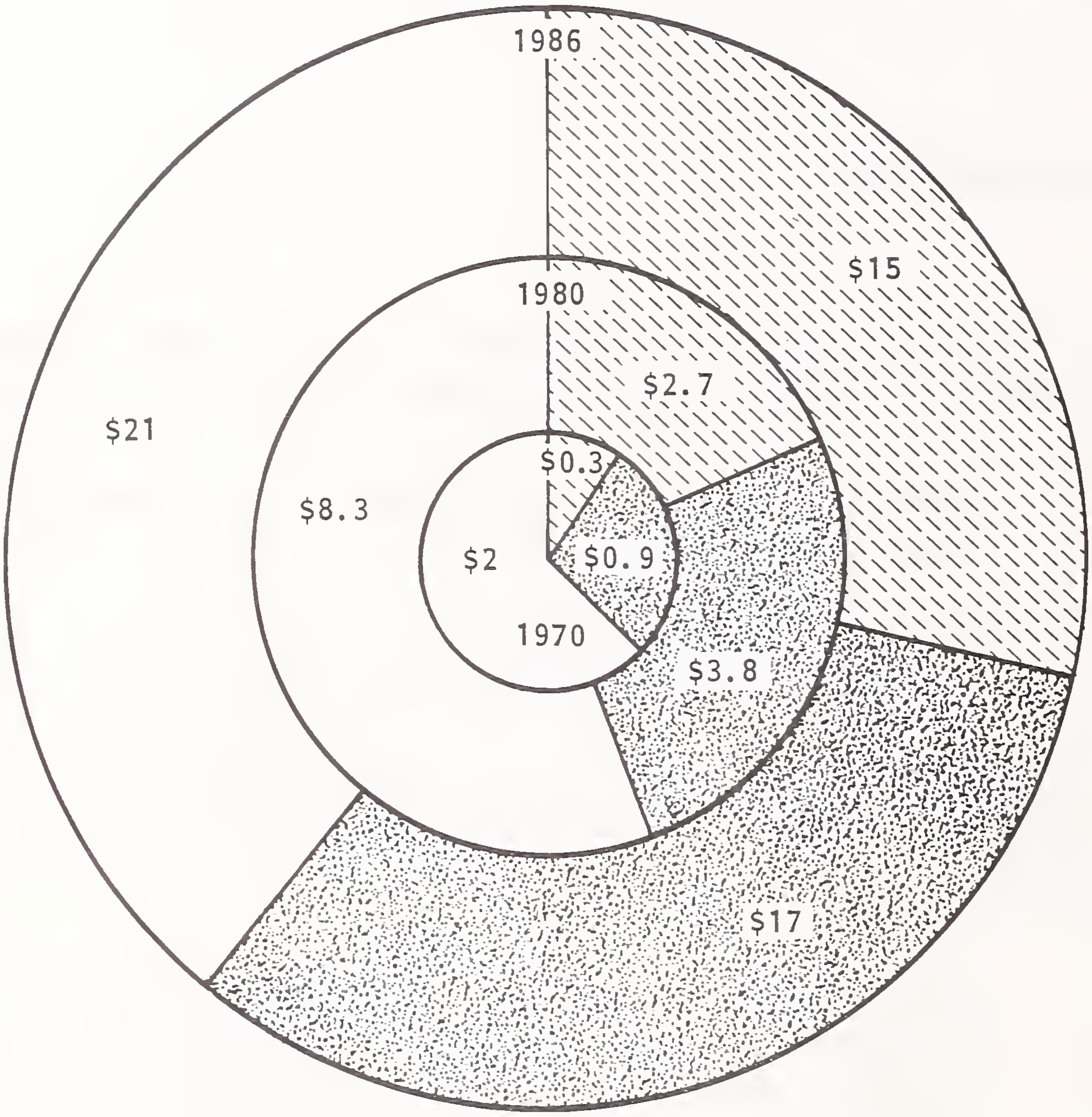
  - Some of the individual industry sectors will be showing very rapid growth as well, as shown in Exhibit I-3.

  - A special case is the growth of software for the very small or personal

EXHIBIT I-1

U.S. INFORMATION SERVICES MARKET, 1970-1986*

($ billions)



MARKET

1970 = $ 3.2 billion
1980 = $14.8 billion
1986 = $53.0 billion

▨ = Software Products

░ = Professional Services

□ = Processing Services

* Does not include weapon system software

EXHIBIT I-2

APPLICATION AND SYSTEMS SOFTWARE PRODUCT GROWTH

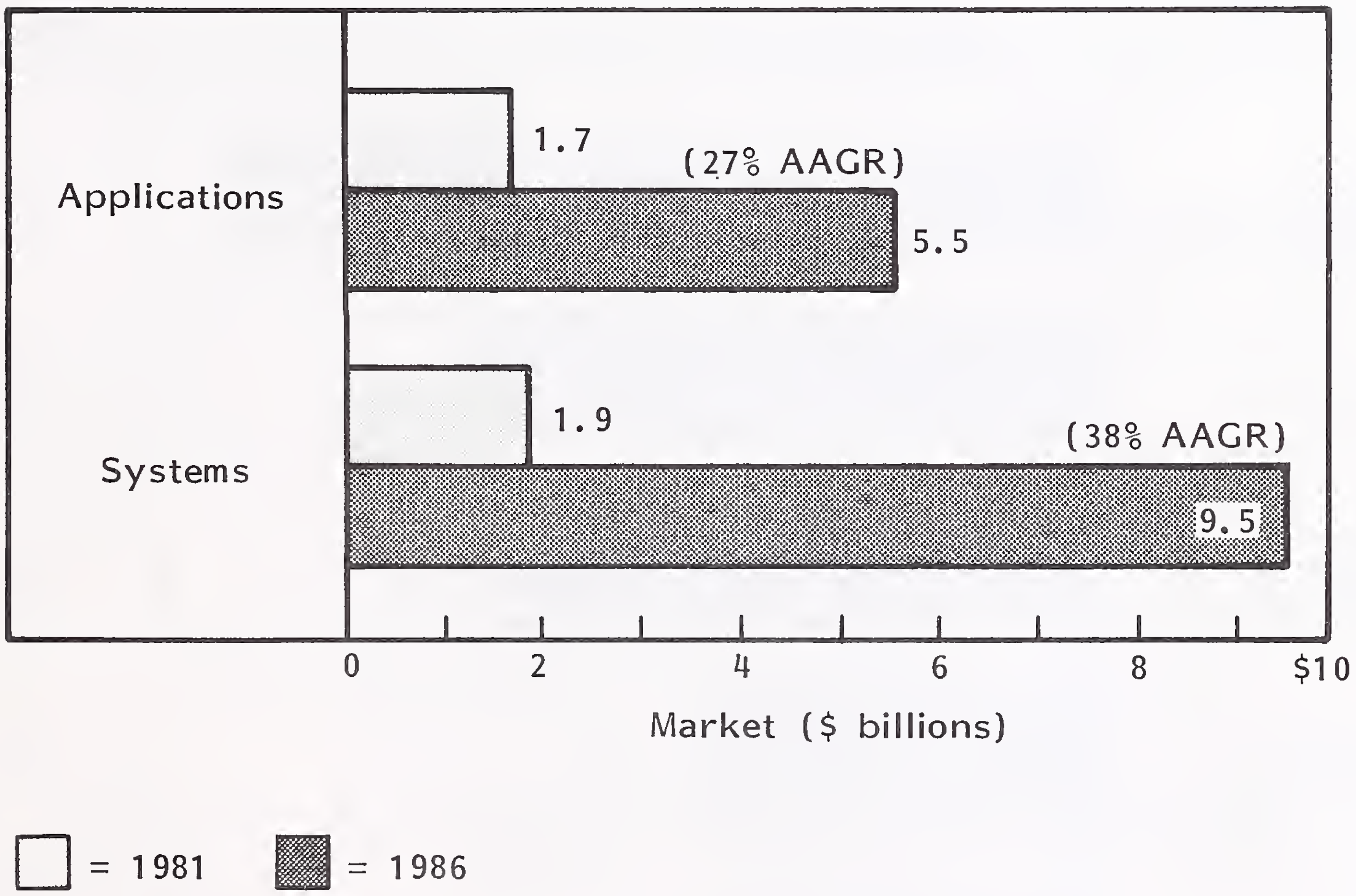
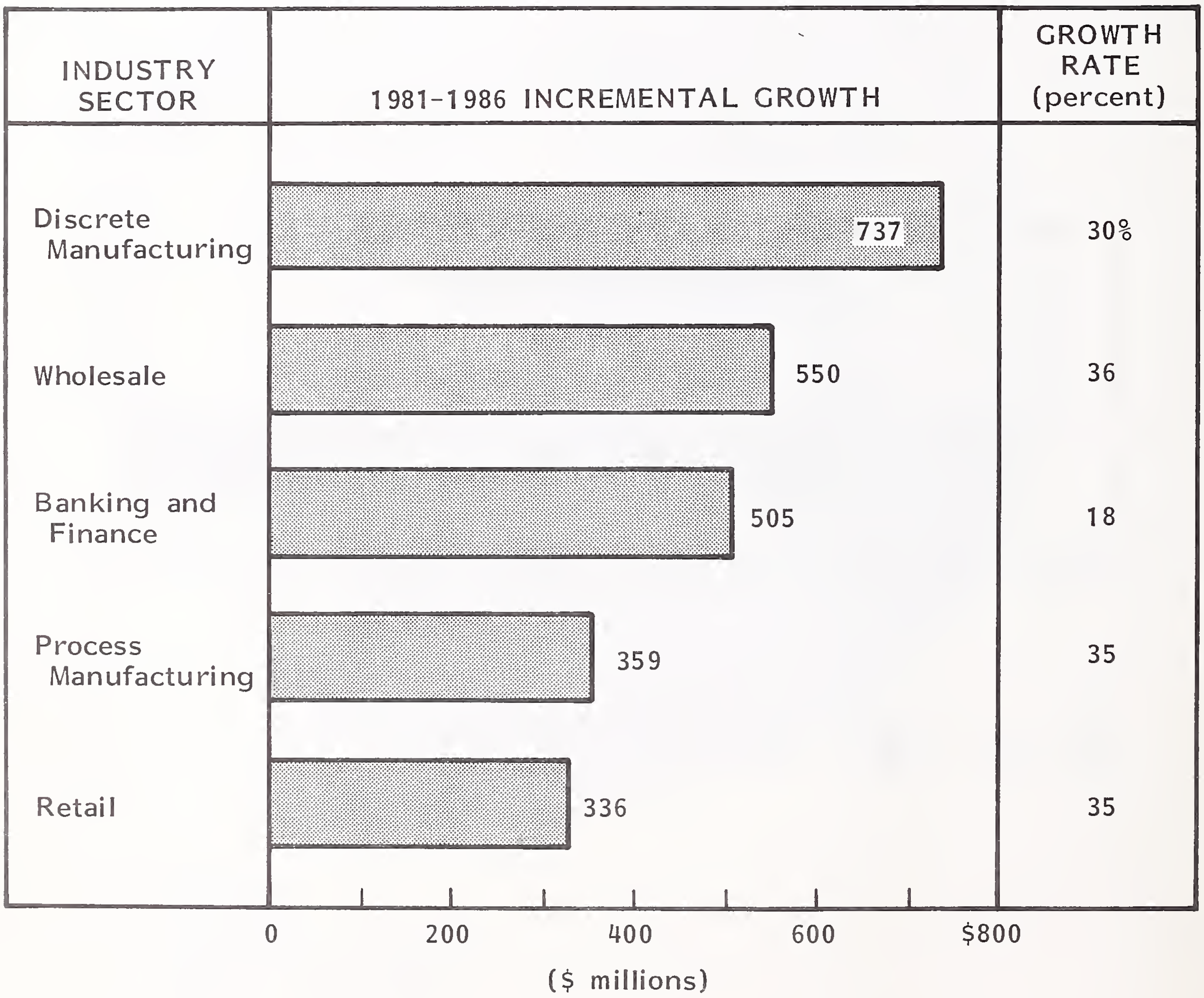


EXHIBIT I-3

APPLICATIONS SOFTWARE PRODUCTS MARKETS - GROWTH IN SELECTED INDUSTRY SECTORS



computer market. Starting from an admittedly small base, this market promises phenomenal growth, as shown in Exhibit I-4.

- Many firms in the industry are beginning to ask themselves how much maintenance of software packages is:

  - A separate, definable product area.

  - A growth area.

  - A profit center.

- Some firms are already exploiting the potential of software maintenance.

  - For example, at a rapidly growing independent software company, Management Science America (MSA), software maintenance revenue is 22% of total revenue and still increasing as a proportion of the total.

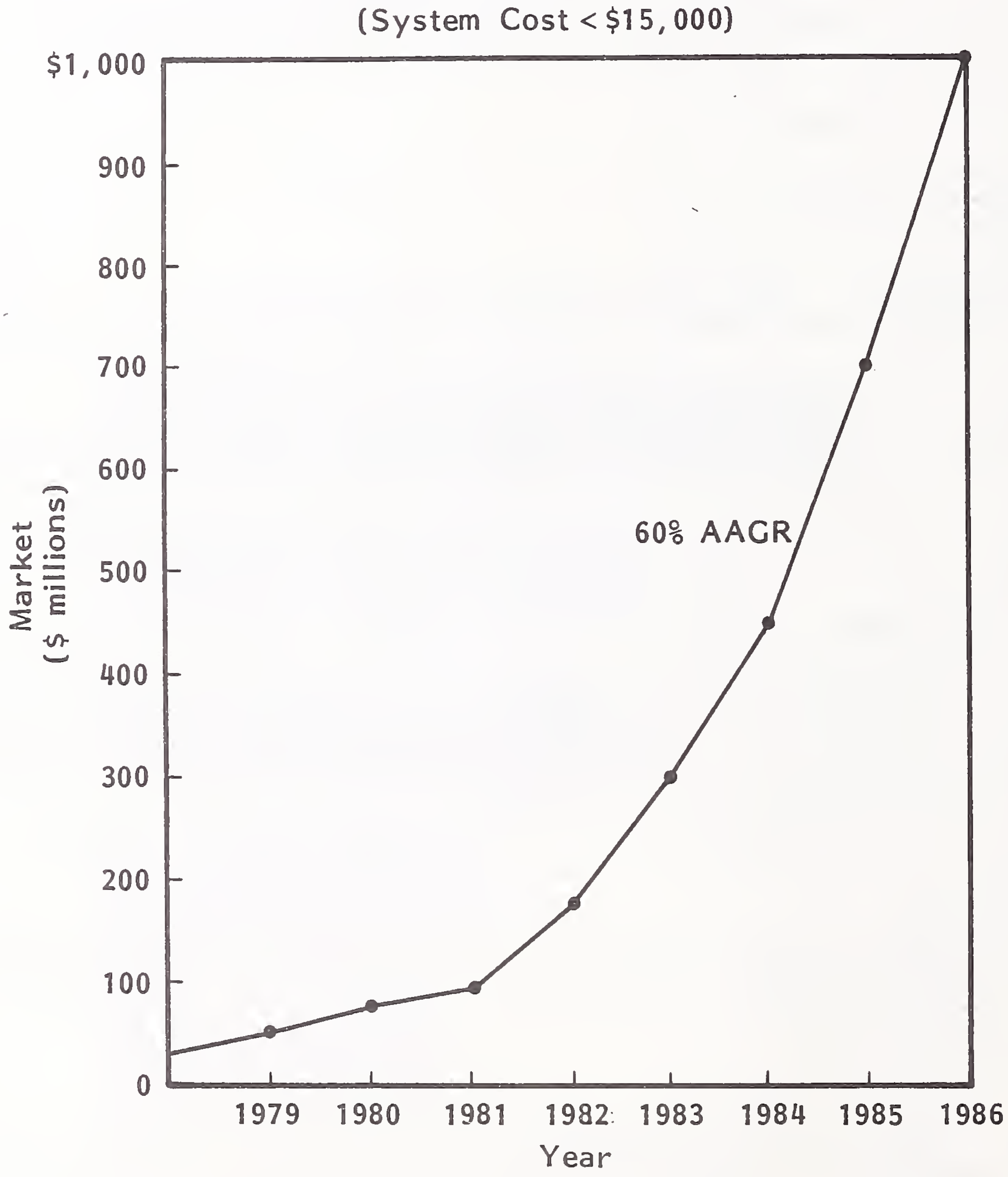
  - The leading firm in supplying packaged software to the property/casualty insurance industry, Policy Management Systems, not only sells its packages at an average price of over $250,000, but also requires customers to sign a five-year maintenance contract with an annual payment of 25% of the original cost of the software.

- On the other hand, customers are scrutinizing many software purchases more and more closely to determine if software really needs vendor-supported maintenance.

- In addition to these external events and trends, many software suppliers are in the midst of evaluating the best way to market and deliver software maintenance. Some companies, for example, are in the process of integrating much of their hardware and software service organizations.

EXHIBIT I-4

VERY SMALL COMPUTER APPLICATIONS PRODUCTS MARKET



- With this background in mind, INPUT determined that a report on software maintenance would be of high interest and utility to field service managers.

B. METHODOLOGY

- INPUT interviewed software marketing and technical management from 37 leading firms in the industry to ascertain current industry practices and future plans.

  - The questionnaire used for this purpose is shown in Appendix A.

  - A description of the firms interviewed is contained in Appendix B.

- Ten of the firms are also significant vendors of hardware. In some cases, their responses differed from vendors who are software-only vendors. To highlight these differences, the responses from hardware and software-only vendors have been segregated and contrasted.

- INPUT also interviewed information systems (IS) management of leading corporations who are members of INPUT's 1982 User Panel to determine their current and planned use of vendor-supplied software maintenance.

- INPUT has also drawn on its knowledge from several special consulting studies in the areas of:

  - Software marketing practices.

  - Software maintenance business opportunities.

  - New business opportunities in computer services.

- IS department organization and mission planning.

C. SCOPE OF THIS REPORT

- This report focuses on the issues of the maintenance of packaged software in the commercial environment.

- The following areas are excluded:

  - Maintaining custom-developed application software. This is a different kind of business from maintaining multiple copies of software.

  - Maintaining other firms' software packages on a third-party basis. In INPUT's view, this is not a feasible business. (For further discussion see Appendix C.)

  - Maintaining weapons systems software. Although large and growing, this business is highly specialized.

D. SOFTWARE PRODUCTS DEFINITIONS

- This category includes the user's purchase of applications and systems packages for use on in-house computer systems. Included are lease and purchase expenditures as well as fees for work performed by the vendor to implement and maintain the package at the user's site(s). Fees for work performed by organizations other than the package vendor are counted in professional services. There are several subcategories of software products:

- <u>Application Products</u> are software products which perform processing to serve user functions. They consist of:

  - <u>Cross-industry products</u>, in multiple-user industry sectors. Examples are payroll, inventory control and financial planning.

  - <u>Industry-specialized products</u>, in a specific industry sector such as banking and finance, transportation, or discrete manufacturing. Examples are demand deposit accounting and airline scheduling.

- <u>System Products</u> are software products which enable the computer/communications system to perform basic functions. They consist of:

  - <u>System operations products</u>, which function during applications program execution to manage the computer system resource. Examples include operating systems, DBMS, communication monitors, emulators, and spoolers.

  - <u>System utilization products</u>, used by operations personnel to utilize the computer system more effectively. Examples include performance measurement, job accounting, computer operations scheduling, and utilities.

II MANAGEMENT SUMMARY

II MANAGEMENT SUMMARY

- Software maintenance will assume increasing importance over the next five years for four reasons.

  - The market will be growing 10 times in that period, even faster than the total software market, as shown in Exhibit II-1.

  - For individual software products, especially those whose growth has stabilized, the software proportion of total revenue can be even more attractive, as shown in Exhibit II-2.

  - Software maintenance represents one of the few areas not subject to cheap "knock-offs", as much of the computing industry moves into the commodity stage.

  - The cost pressures on supplying software maintenance will increase, since so many present activities are labor-intensive. Companies that increase software maintenance productivity will prosper.

- Software products themselves can be highly leveraged. Software maintenance and support at this time are not. Methods of improving leverage include:

  - Innovative interactive training methodologies, useful for both installation training and as a replacement for labor-intensive and variable quality hotline services.

EXHIBIT II-1

SOFTWARE AND SOFTWARE MAINTENANCE REVENUE: 1980-1986

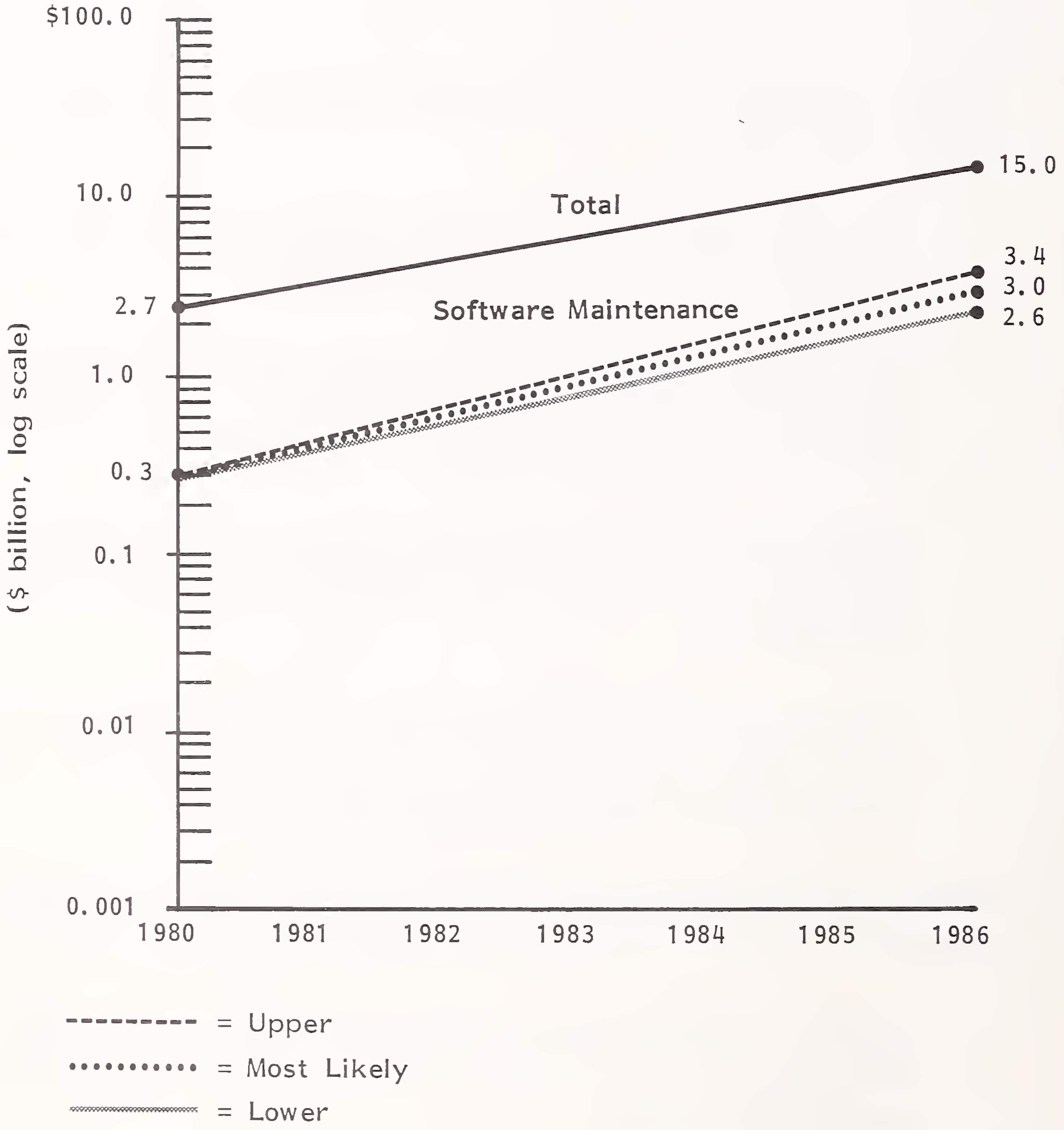
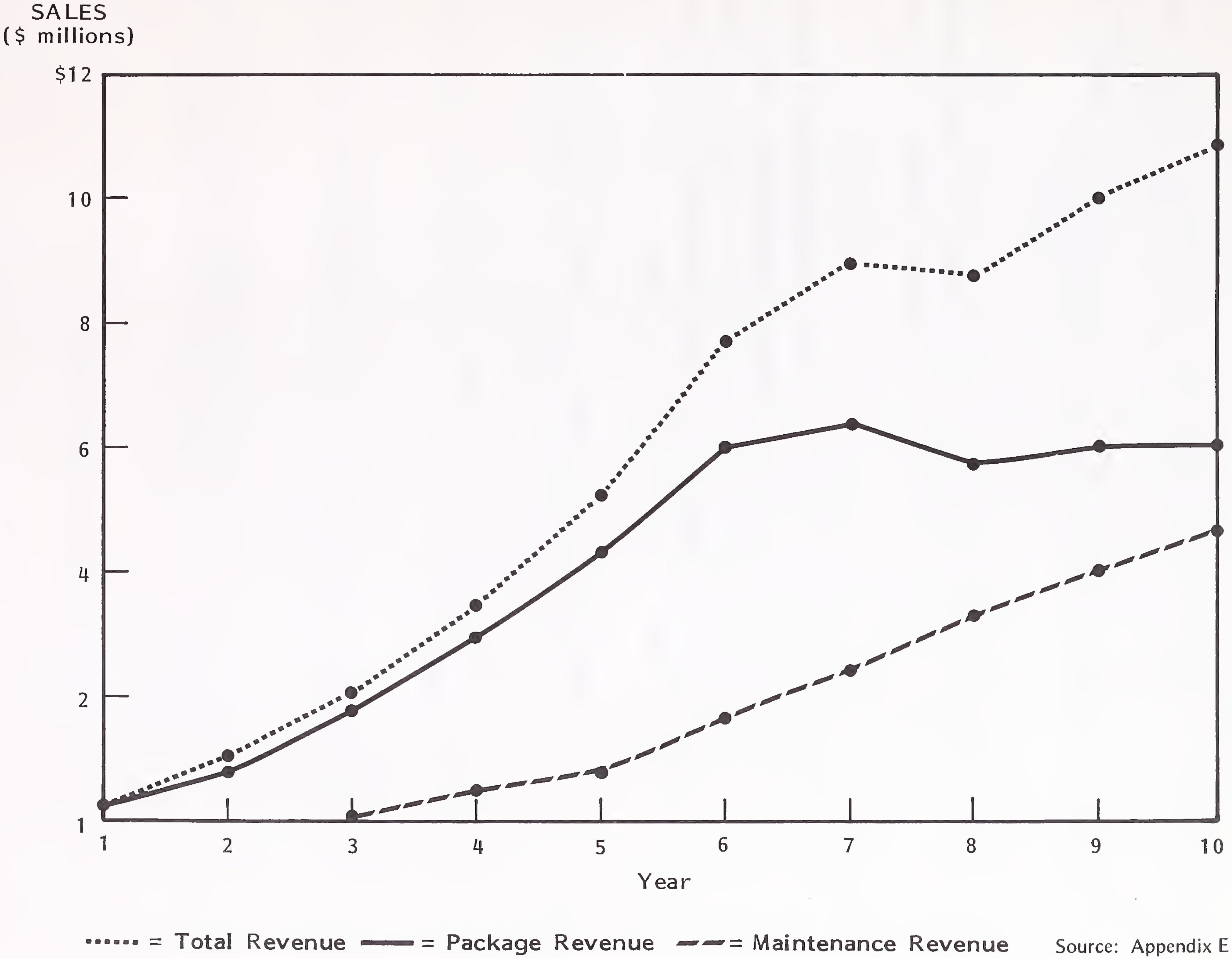


EXHIBIT II-2

REVENUE COMPOSITION FOR A HYPOTHETICAL SOFTWARE PRODUCT



Source: Appendix E

- Integrated software development and maintenance tools. These will require years of waiting before they have an impact on current maintenance operations. However, companies that do not make the investment now will fall behind in the critical years ahead.

- Some of the lessons learned in the hardware maintenance business will not be applicable to software maintenance. Integral parts of software maintenance are making product improvements and supplying training and consulting on how to use the product.

- Currently, the market is fairly insensitive to price; in addition, much of the market is captive, especially for hardware-unique software. This will change in the future due to cheap software on one end of the applications market and operating systems competition in all parts of the market, including IBM mainframes. Vendors can respond to this in two ways:

  - Consciously structuring value into software maintenance offerings by providing more specificity in products.

  - Unbundling software maintenance into its constituents and pricing each appropriately.

- Many vendors expect to increase software maintenance revenue significantly while holding steady or decreasing costs.

  - This will not be feasible whether existing maintenance processes are followed (because of ongoing expense) or new technologies are implemented (because of startup costs).

  - Vendors should review such decisions.

- Many vendors are reviewing the organizational placement of software maintenance: does it belong in marketing (where it usually is) or should it be moved to field service?

  - There are advantages and disadvantages to either location. The deciding points are usually company-unique.

  - The organizational placement is almost always less important that the strategy and plans for carrying it out.

- One organizational move is desirable in itself: making the central maintenance unit independent of the development organization. This is not, however, always feasible.

III SOFTWARE MAINTENANCE PRACTICES AND ISSUES

III SOFTWARE MAINTENANCE PRACTICES AND ISSUES

A. OVERVIEW

- This chapter begins by defining the boundaries of software maintenance in the vendor marketplace.

- The remaining sections address:

  - Pricing.

  - Resource allocation (budgeting).

  - Marketing and sales.

  - Distribution methods.

B. WHAT IS SOFTWARE MAINTENANCE ?

- "Software maintenance" does not have a commonly accepted definition in either the user or vendor communities.

  - Information systems departments have elastic definitions of maintenance when maintaining their own in-house developed software: maintenance covers functions ranging from fixing minor bugs to system rewrites encompassing many man-years of effort.

  - This confusion carries over into vendor activities. It is at least partly influenced by the lack of clarity of IS departments' expectations.

- Virtually all vendors agree that fixing software errors is included in software maintenance, as shown in Exhibit III-1. It is interesting that a few software vendors do not see even this as part of their responsibilities.

  - Most vendors also see improving, adding, and extending features as part of software maintenance.

  - Software vendors are much less likely than hardware vendors to include training and consulting in maintenance.

  - Supplying conversion and interface assistance are seen by only a minority of vendors as being part of maintenance.

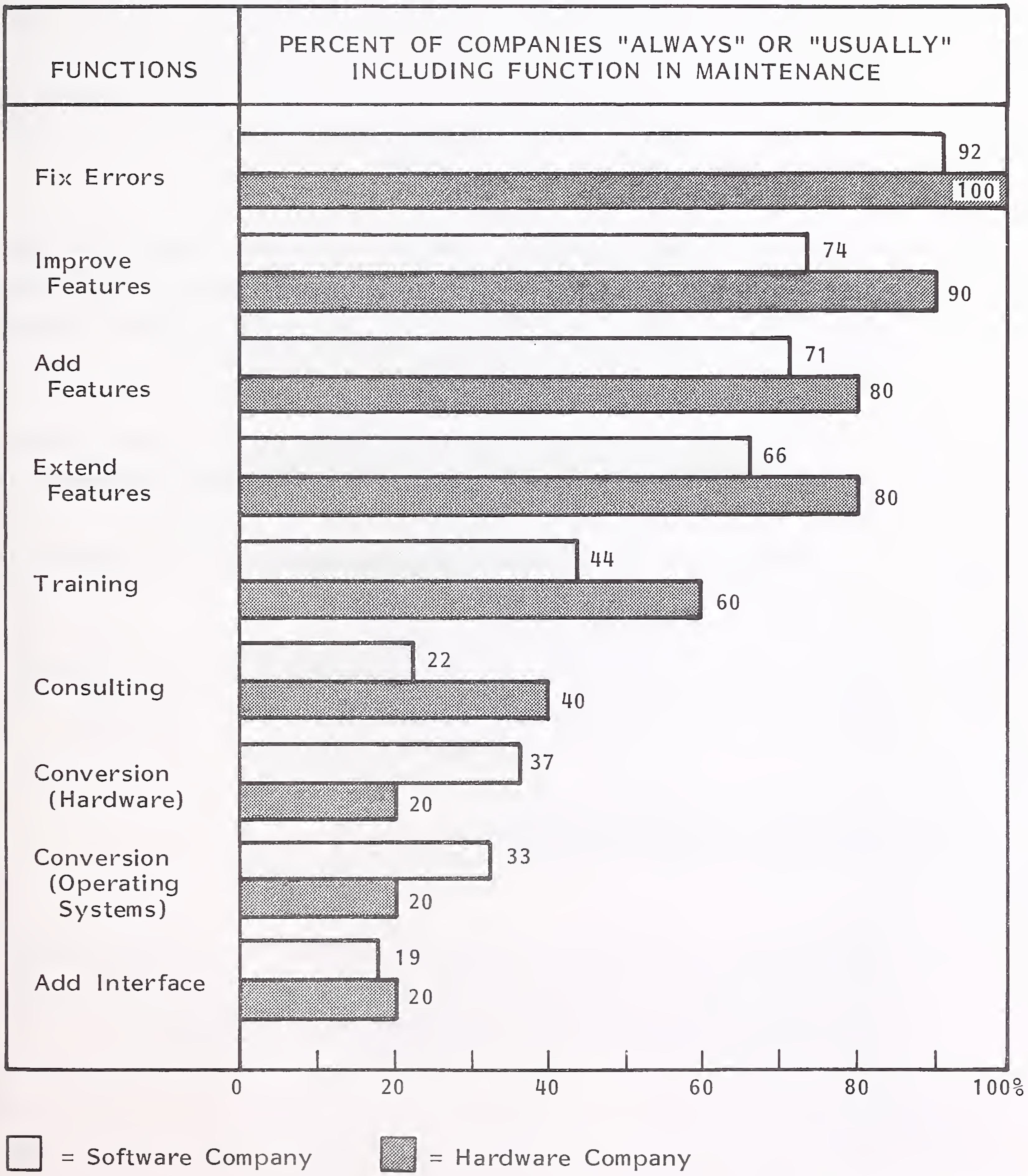
  - Generally, software vendors include fewer activities in maintenance than hardware vendors, except for conversions.

    - Hardware vendors take a more inclusive view of maintenance because they are used to taking a more comprehensive view of customers' needs; in addition a bundled services attitude in many cases has survived unbundling.

    - The exception for conversions points up the different roles of hardware and software companies. Hardware companies will only consider conversions within their own hardware line, while software companies will make any conversions that are economically attractive.

EXHIBIT III-1

FUNCTIONS INCLUDED IN VENDOR MAINTENANCE SOFTWARE



SOURCE: INPUT Survey

- Hardware vendors have not changed their definition of maintenance in the past three years. However, 30% of the software vendors reported doing so to adapt to new markets and product areas.

- Both hardware vendors (60%) and software vendors (44%) expect to be making changes in the activities included in software maintenance. Both types of vendor will try to reduce the extent of services and activities included in maintenance, as part of their efforts to reduce the resources and costs of software maintenance.

- It is noteworthy that while fewer than half the vendors view training and consulting as activities normally part of software maintenance, 60% of vendors see dealing with user misuse or lack of understanding as the key maintenance activity, as shown in Exhibit III-2.

  - Error correction accounts for only 13% of activities. (Note: this is within the 10-20% range commonly reported for in-house maintenance.)

  - Technology issues (e.g., conversions, upgrades, or improved efficiency) account for less than one-fifth of activities.

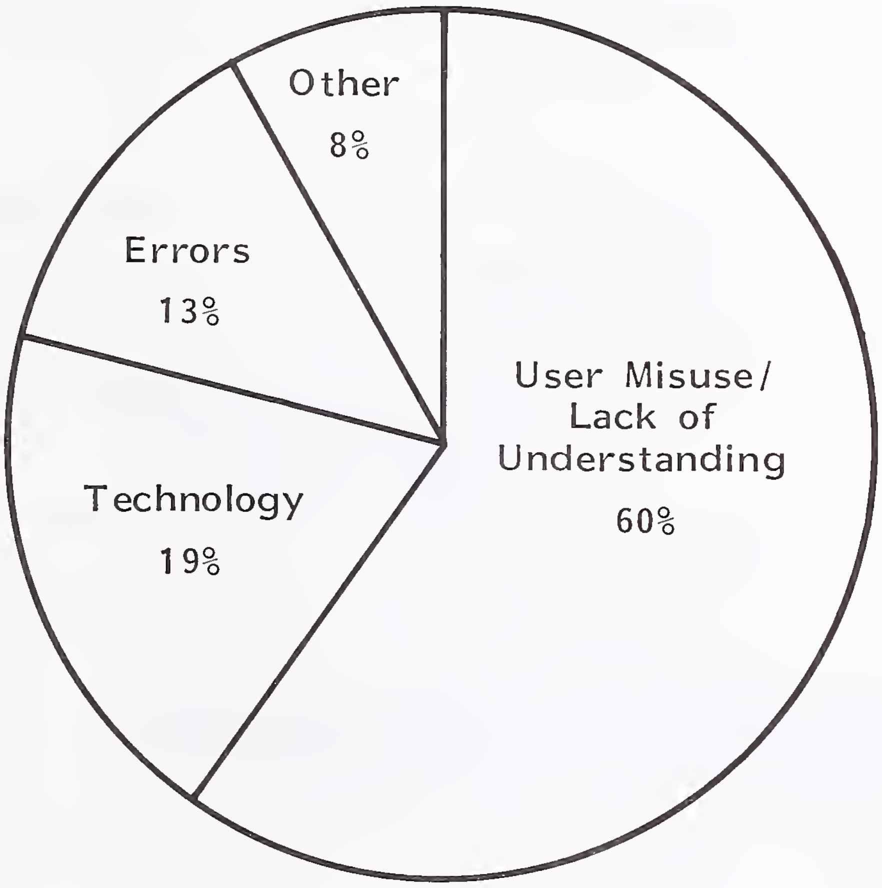
- There is consequently a built-in tension between what vendors see as software maintenance and the actual demands on the software maintenance area.

C. SOFTWARE MAINTENANCE PRICING

- Vendors' current estimates for the proportion of their software revenue which comes from software maintenance ranges from 4% to 50% (vendors who still bundle their software or software maintenance are not included).

EXHIBIT III-2

FREQUENCY OF MAINTENANCE ACTIVITIES



SOURCE: INPUT Survey

- The maintenance portion of software vendors' revenue is 15%, while the maintenance portion of hardware vendors' software revenue is 9%, as shown in Exhibit III-3.

- Software vendors see a modest growth in this proportion over the next five years, while hardware vendors see the software maintenance share increasing by a factor of four.

  . Some of this increase is accounted for by further unbundling and price increases to captive customers.

  . However, INPUT believes that some of the expectations for revenue increases of this magnitude are not realistic, unless sales of additional software units would fall off drastically; this is certainly not what hardware companies have in mind.

  . Consequently, INPUT believes that hardware vendors will be fortunate if they can match the software vendor's 20% figure.

● There is certainly room for justified increases in software and software maintenance prices. INPUT's ongoing custom research in this area has shown that, across industry and product groups, price is not now a major consideration for most customers.

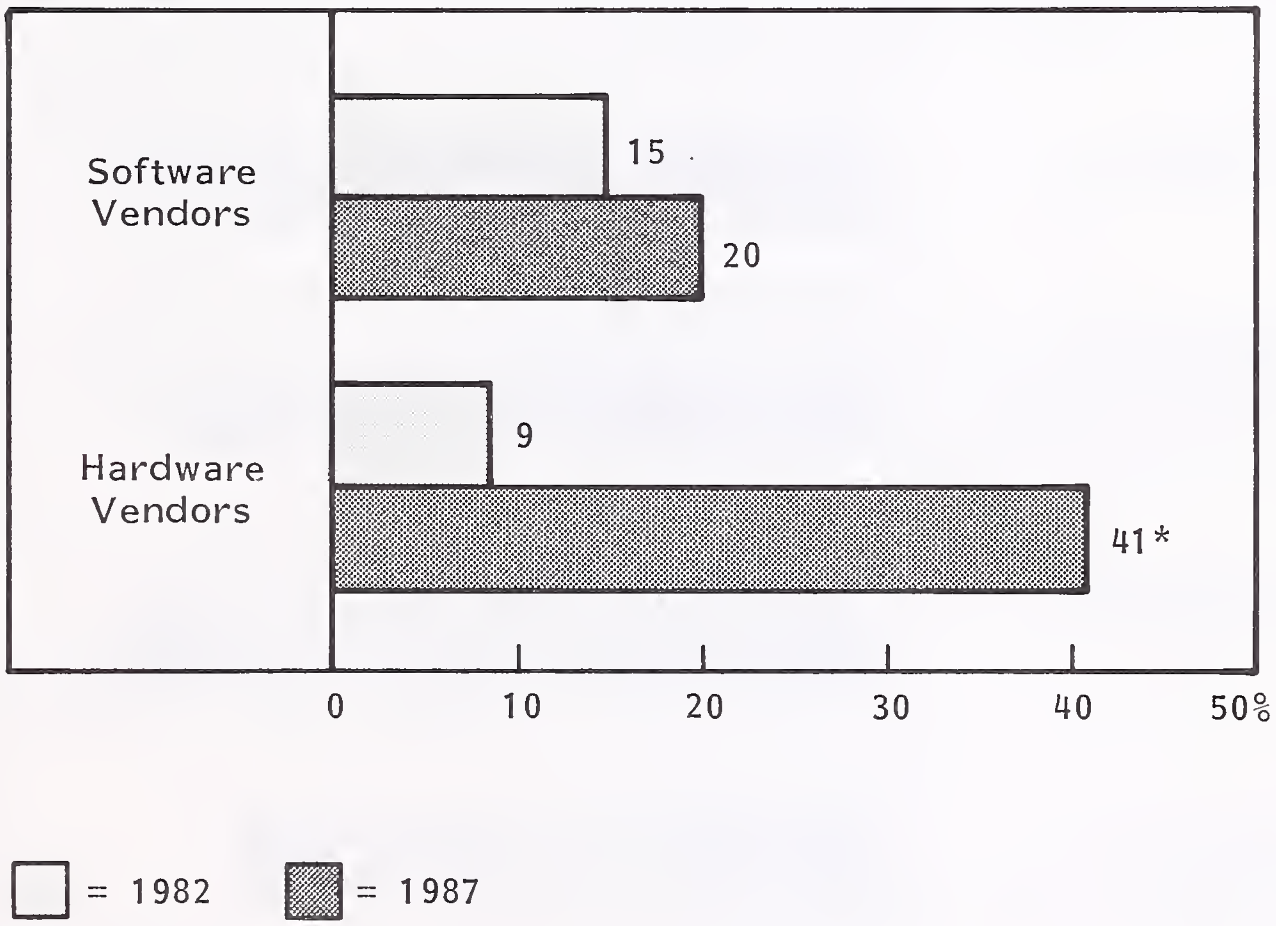
  - Customers' high priorities are for functionality, flexibility, and support. Customers will buy a software product that they perceive to be overpriced (from a supplier cost/profit standpoint) if it meets these needs better than competing products.

  - Vendors typically ascribe more importance to price than customers do.

● In general, vendors say that similar importance is given to each of the factors in pricing software maintenance shown in Exhibit III-4:

EXHIBIT III-3

VENDOR FORECASTS OF PROPORTION OF SOFTWARE REVENUE COMING FROM SOFTWARE MAINTENANCE

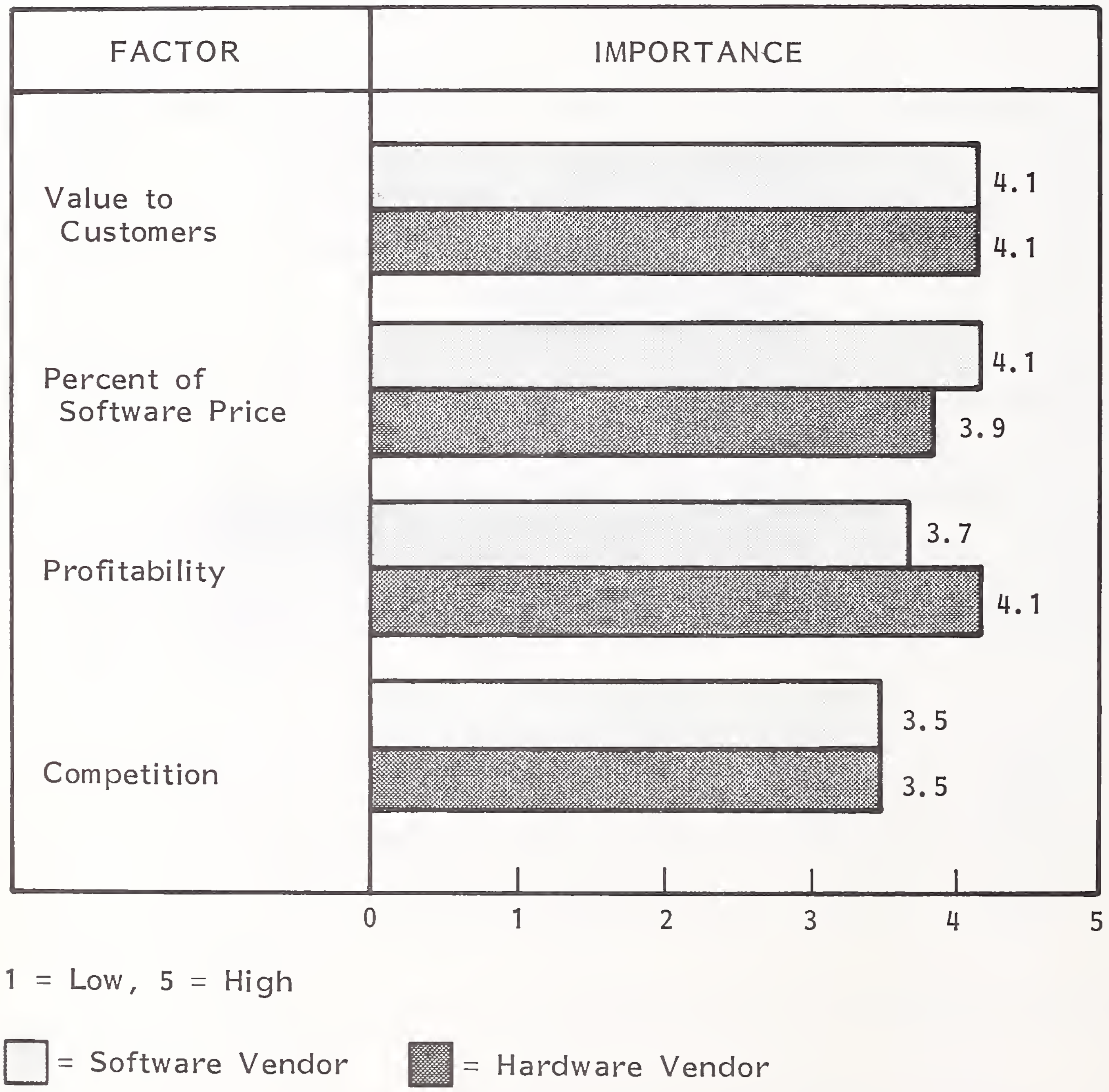


* INPUT believes that 20% is more realistic.

SOURCE: INPUT Survey

EXHIBIT III-4

IMPORTANCE OF FACTORS IN DETERMINING SOFTWARE MAINTENANCE PRICING



SOURCE: INPUT Survey

- Value to customers.

- Percent of software price.

- Profitability.

- Competition (industry norm).

- However, 84% of the companies interviewed only used one method to determine pricing for software maintenance. Most companies use a mechanistic approach to pricing, either a percent of the package price or a profitability target, as shows in Exhibit III-5. This means that maintenance pricing may be too low or too high.

    - Pricing too low leaves money on the table.

    - Pricing too high may cause some customers to avoid vendor maintenance, thereby possibly reducing total software maintenance revenue. This may cause even more serious longer-run problems, as analyzed in Chapter V.

D. RESOURCE ALLOCATION

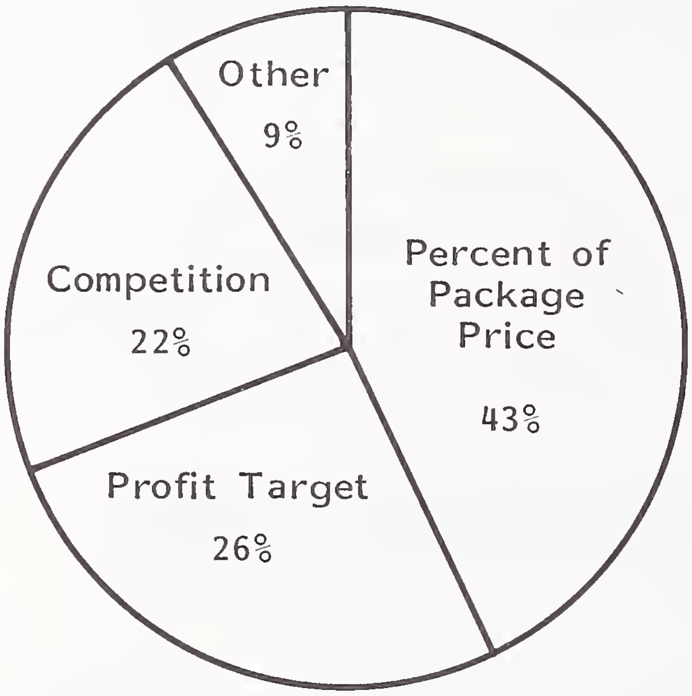
- Almost half the vendors interviewed (48%) use a formal budget process to allocate resources for software maintenance.

    - The 44% of software vendors which use a budgeting process are generally satisfied with it.

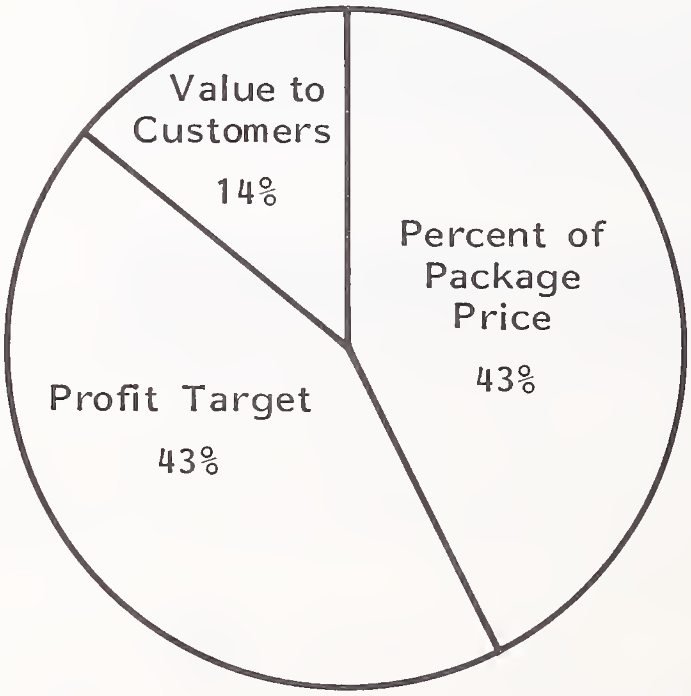
    - While 60% of hardware vendors use a budgeting process, they are less satisfied than the software vendors. This appears related to the

EXHIBIT III-5

METHODS OF DETERMINING
SOFTWARE MAINTENANCE PRICING



SOFTWARE COMPANIES



HARDWARE COMPANIES

SOURCE: INPUT Survey

greater number of competing demands for resources within a hardware company.

- Different methods are used to allocate resources, as shown in Exhibit III-6. The main methods are:

  - A percent or ratio applied to the overall software budget or against the total number of customers.

  - A historical method, modifying the prior year's budget upward or downward.

  - Projecting future requirements.

- Some companies use more than one method. All methods are somewhat arbitrary given the difficulty of predicting what changes will be necessary.

  - Some companies attempt to deal with this by planning to introduce in the course of a planning cycle:

    - X new products.

    - Y major revisions.

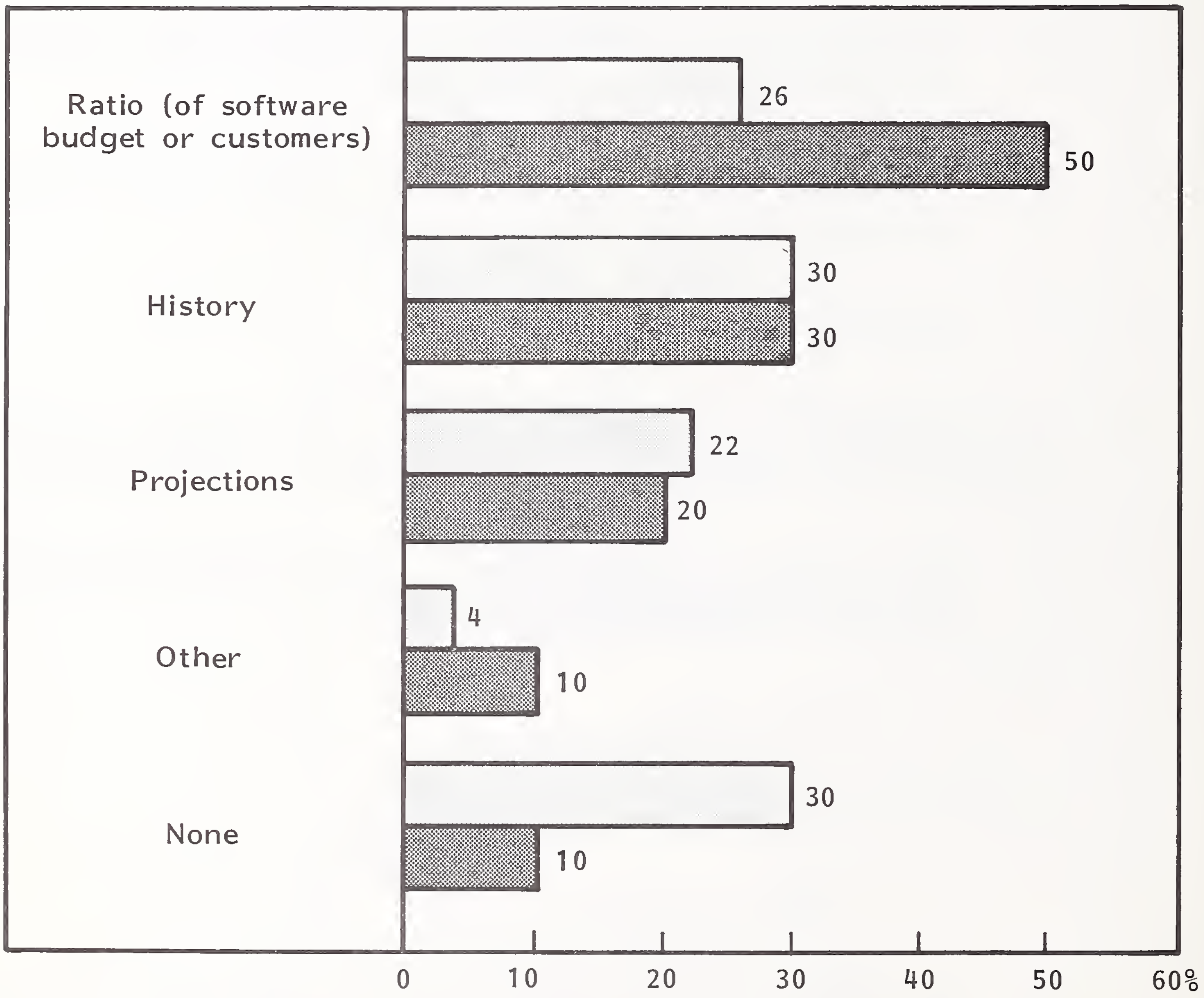
    - Z minor revisions.

  - This logical approach can usually yield at least ballpark cost estimates. However, it may result in unacceptable lead times and larger or smaller changes than the market needs.

- Resource allocations are not immutable (nor should they be). Three-quarters of respondents report shifting resources between the development and maintenance areas.

EXHIBIT III-6

METHODS OF ALLOCATING RESOURCES TO SOFTWARE MAINTENANCE



NOTE: Totals are more than 100% because some firms use multiple approaches.

□ = Software Vendors ▨ = Hardware Vendors

SOURCE: INPUT Survey

- Maintenance personnel represent an emergency source of resources for development (and vice-versa).

- A major product enhancement can be an addition to an existing product, or a new product. (The strategic question of choosing the correct alternative is discussed in Chapter II, Section B-1).

- While this flexibility is useful it can interfere with personnel and product planning. Equally important, it can undermine the software maintenance function's rationale and organizational standing. (These important organizational issues are dealt with in Chapter IV, Section A.)

● While the percent of software maintenance revenue is expected to increase, as shown in Exhibit III-3, few companies expect to raise the relative level of resources devoted to software maintenance, as shown in Exhibit III-7. In fact, more companies expect decreases than increases.

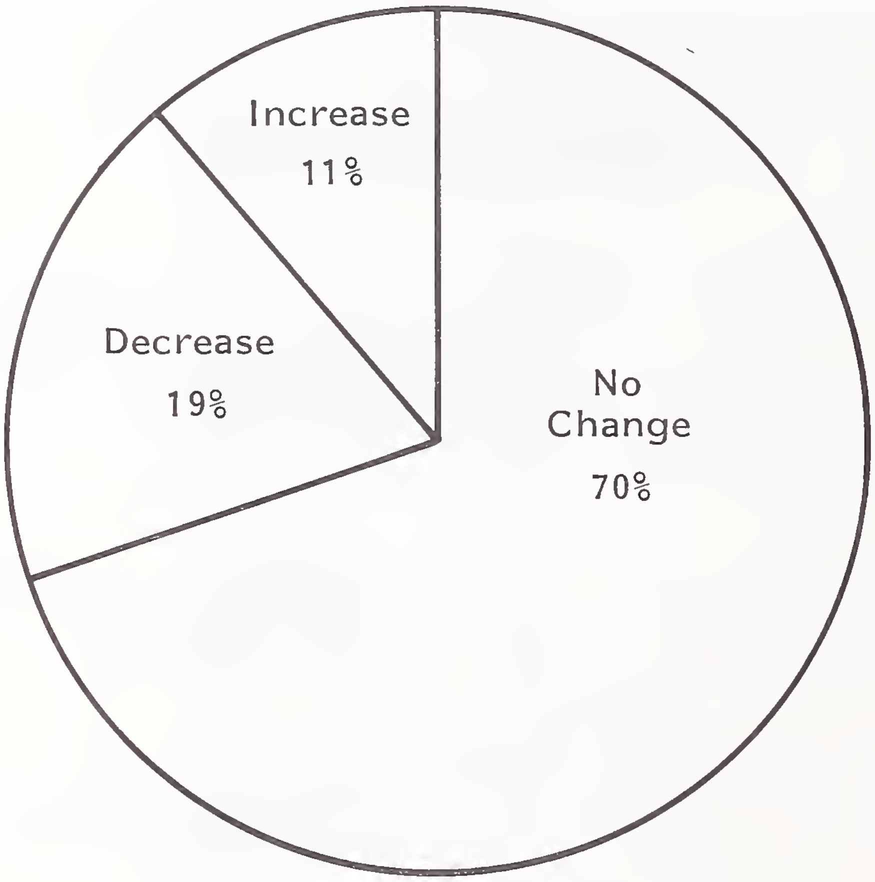
● Resource levels change for different reasons, not just a single reason, such as plans to make software a profit and loss (P&L) center.

  - About 30% of the companies interviewed now have software maintenance as a separate P&L, as shown in Exhibit III-8. Another 25% had plans to do so in the future.

  - The feasibility of making software maintenance a P&L center will vary from company to company, depending on how much control it can exercise over costs, products, and revenue.

EXHIBIT III-7

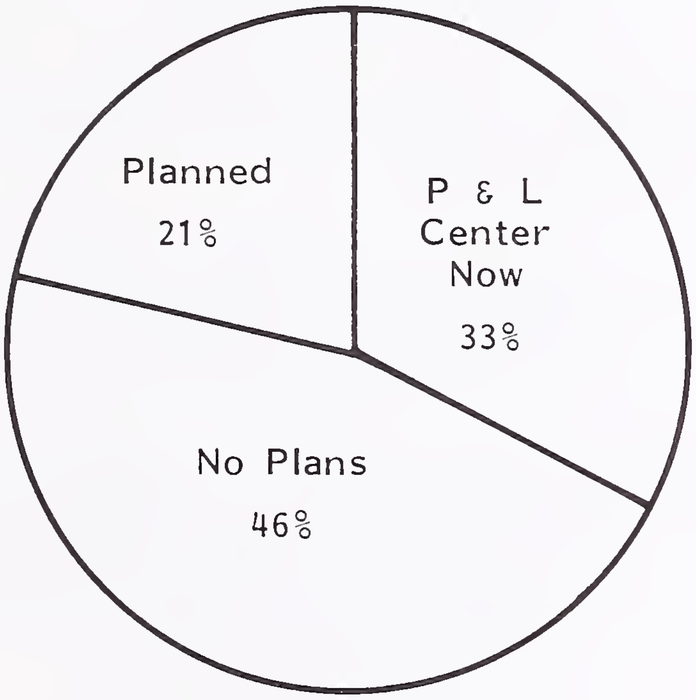
CHANGES SEEN IN LEVEL OF RESOURCES
COMMITTED TO SOFTWARE MAINTENANCE



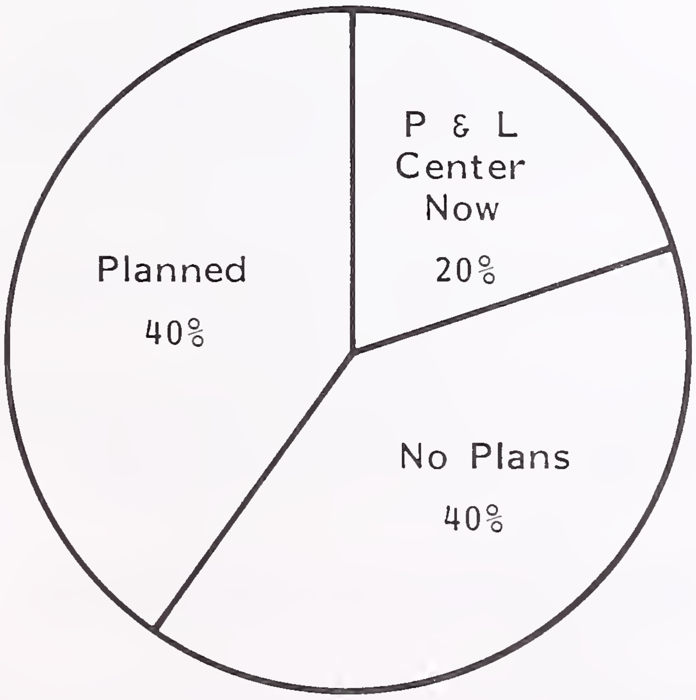
SOURCE: INPUT Survey

EXHIBIT III-8

SOFTWARE MAINTENANCE AS A P & L CENTER



SOFTWARE VENDORS



HARDWARE VENDORS

SOURCE: INPUT Survey

E. MARKETING AND SALES

- Software companies believe their customers are more satisfied than do hardware companies, as shown in Exhibit III-9.

    - Software company customers are rarely captives, as many customers of hardware companies are.

    - Software companies do not have to offer and support the range of software of many hardware companies.

    - Software companies, generally younger and smaller, can be more responsive to customers. However, software companies often suffer from growing pains, which can inhibit a satisfactory customer service effort.

- Even companies with satisfied customers today can have unsatisfied customers tomorrow without adequate information on customer needs and problems.

    - Vendors generally use all means of identifying software maintenance needs, as shown in Exhibit III-10.

        - Problem analysis reports and hotlines are most important.

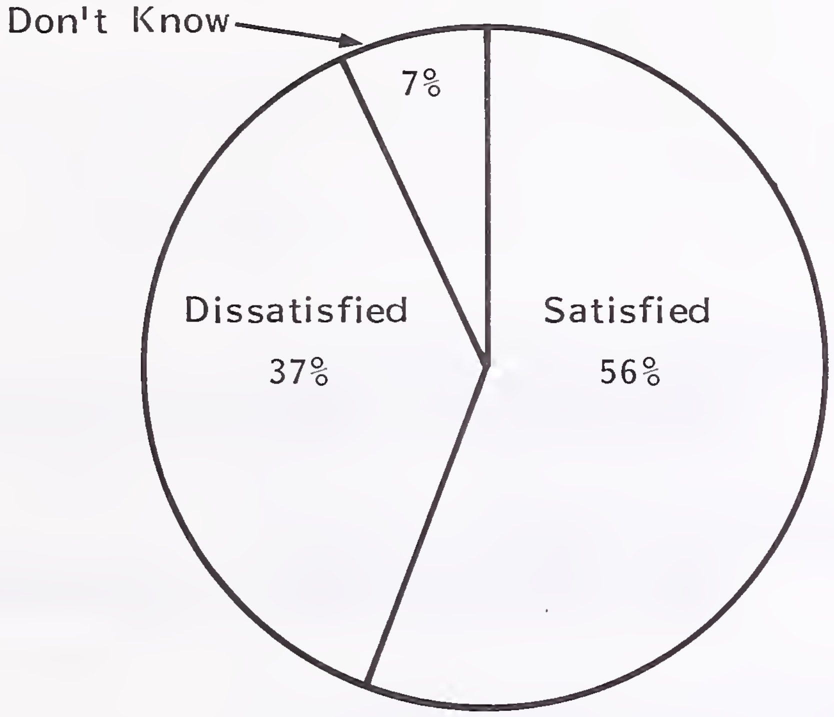
        - Support staff feedback, user groups, field visits, and sales force feedback are almost as important.

        - Surveys are somewhat less important.

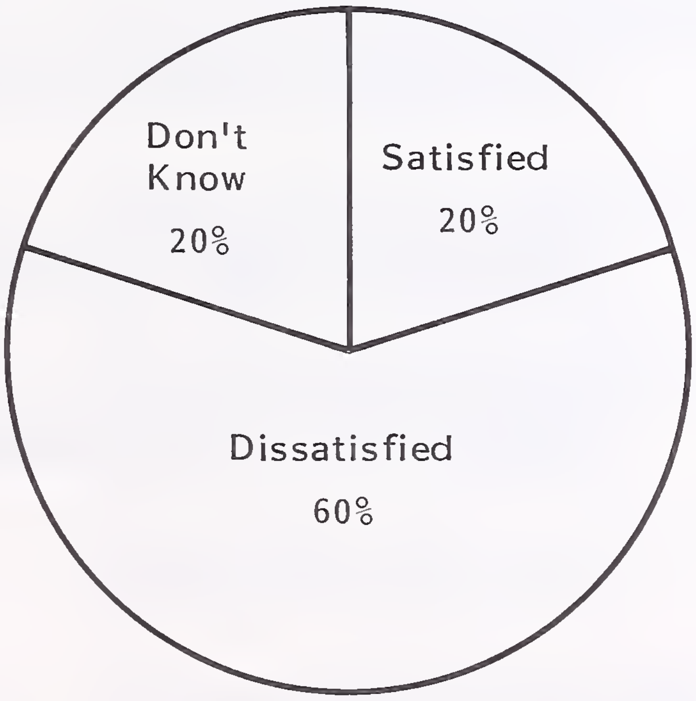
    - The differences between software and hardware companies are minimal.

EXHIBIT III-9

CUSTOMER SATISFACTION WITH SOFTWARE MAINTENANCE



SOFTWARE COMPANIES



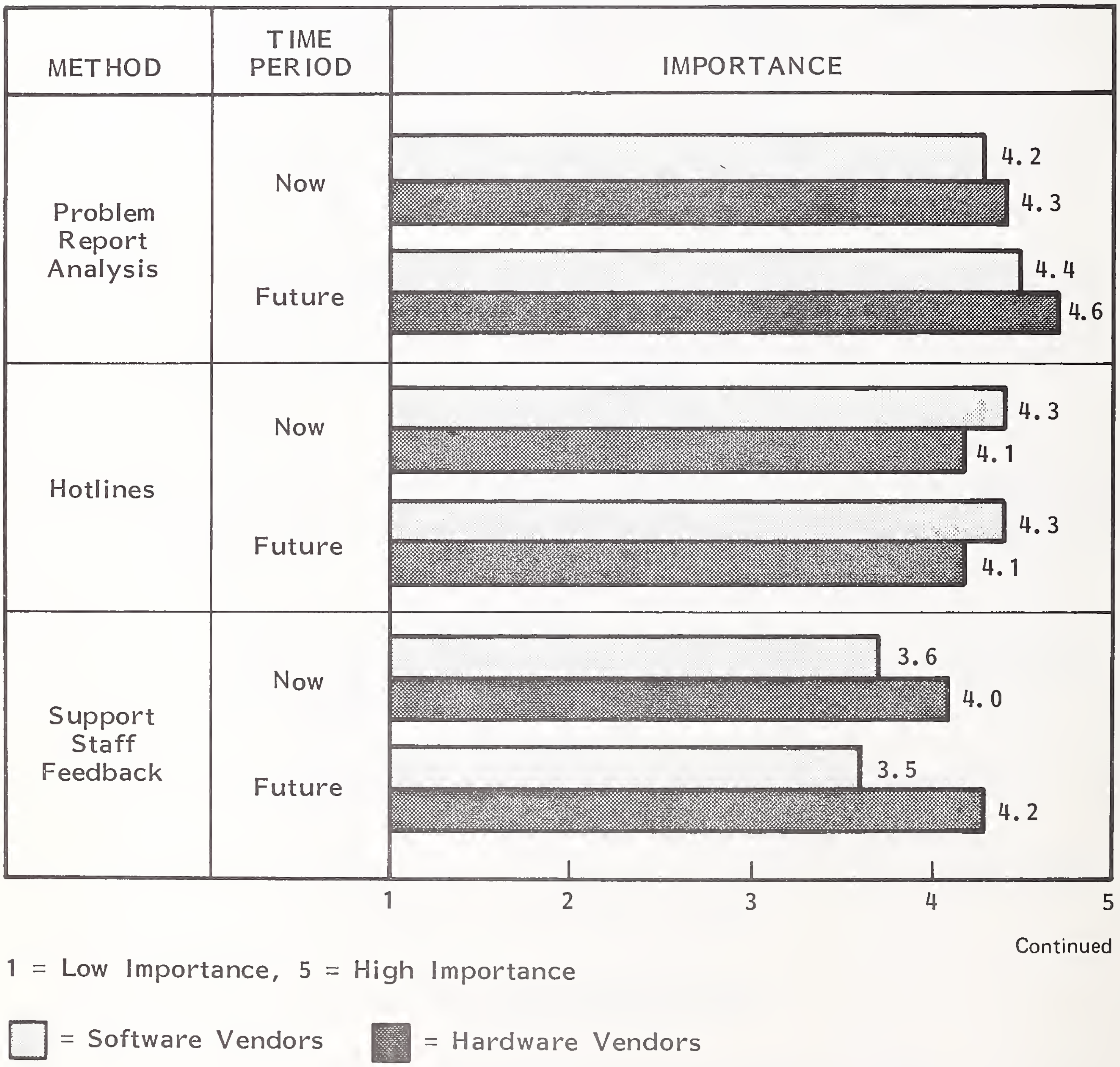
HARDWARE COMPANIES

Percent of Companies Perceiving
Their Customers as Satisfied

SOURCE: Input Survey

EXHIBIT III-10

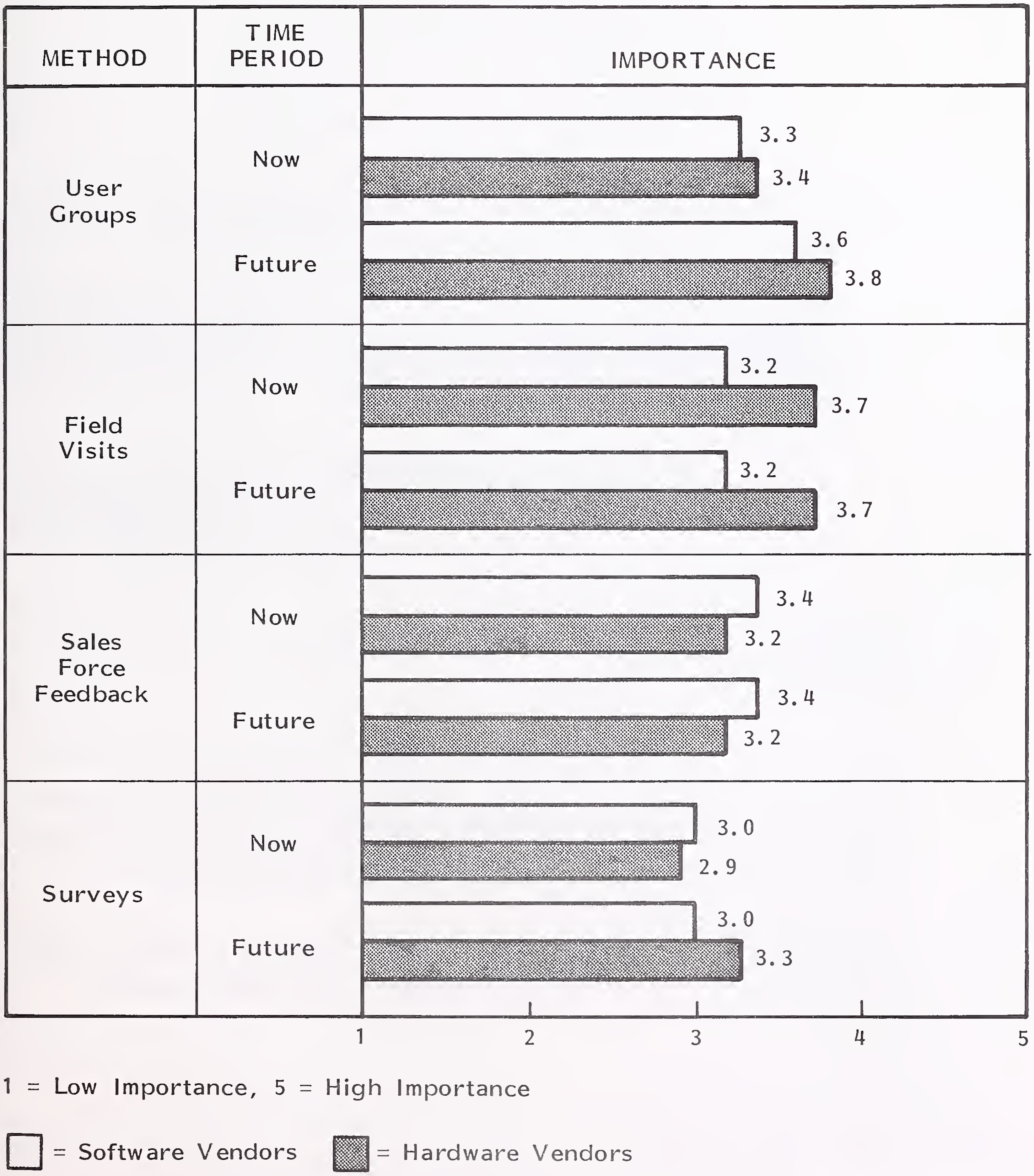
IMPORTANCE OF METHODS OF IDENTIFYING SOFTWARE MAINTENANCE NEEDS



Continued

EXHIBIT III-10 (Cont.)

IMPORTANCE OF METHODS OF IDENTIFYING SOFTWARE MAINTENANCE NEEDS



SOURCE: INPUT Survey

- Respondents see the future as much like the past:

  - Problem reports and user groups will increase slightly in importance for all companies.

  - Hardware companies will rely more on surveys.

  - Other methods will remain about the same.

- Most vendors now perceive software maintenance sales as almost automatic, needing little initiative on their part, as shown in Exhibit III-11.

  - However, many software vendors, but no hardware vendors, believe that more active selling will be required in the future, as shown in Exhibit III-11.

  - Hardware vendors believe that their software maintenance markets will continue to be as protected as they are now. This may not be so; the factors affecting this are analyzed in Chapter V.

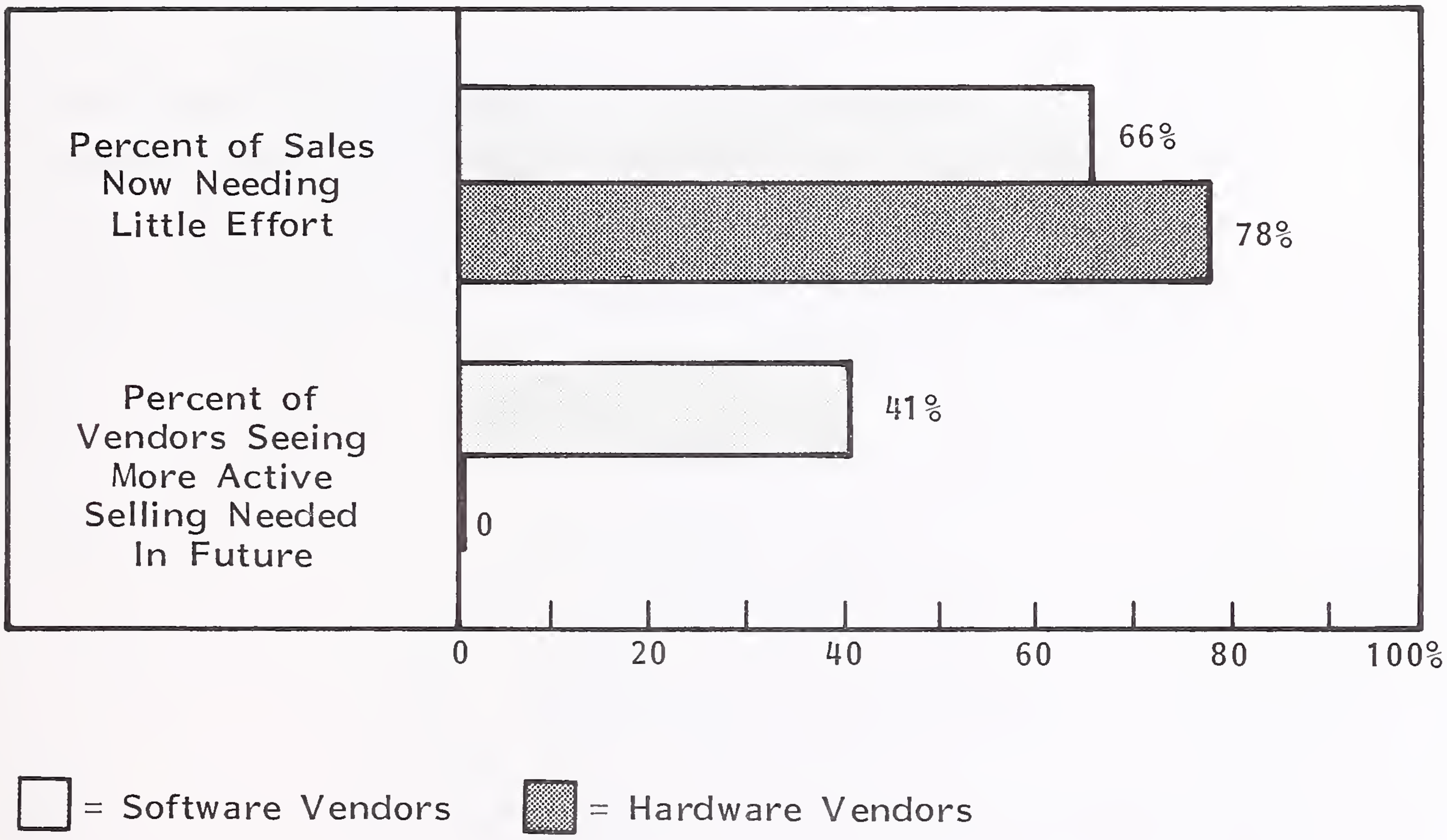
F. DISTRIBUTION METHODS

- Distribution of software revisions is the core and culmination of software maintenance activities. Every revision is expensive to produce and creates additional expense as users adjust to the new software environment.

- It is significant that software vendors appear to have less expensive, more effective software revision distribution methods than hardware vendors:

EXHIBIT III-11

SOFTWARE MAINTENANCE SALES EFFORT



SOURCE: INPUT Survey

- Only 11% of software vendors average three or more software product revisions annually, compared to 40% of hardware vendors, as shown in Exhibit III-12.

- Virtually all software vendors interviewed (84%) had 90% or more of their revisions installed by their customers, as shown in Exhibit III-13.

- Three-quarters of software vendors believed that all or almost all software problems were resolved in the course of the regular software revision cycle, compared to half as many hardware vendors, as shown in Exhibit III-14.

● This superior performance by software vendors occurs even though they are much less automated in their distribution methods than hardware vendors.

- As shown in Exhibit III-15, fewer than 10% of software vendors use telecommunications for:

  . Identifying software problems.

  . Solving software problems.

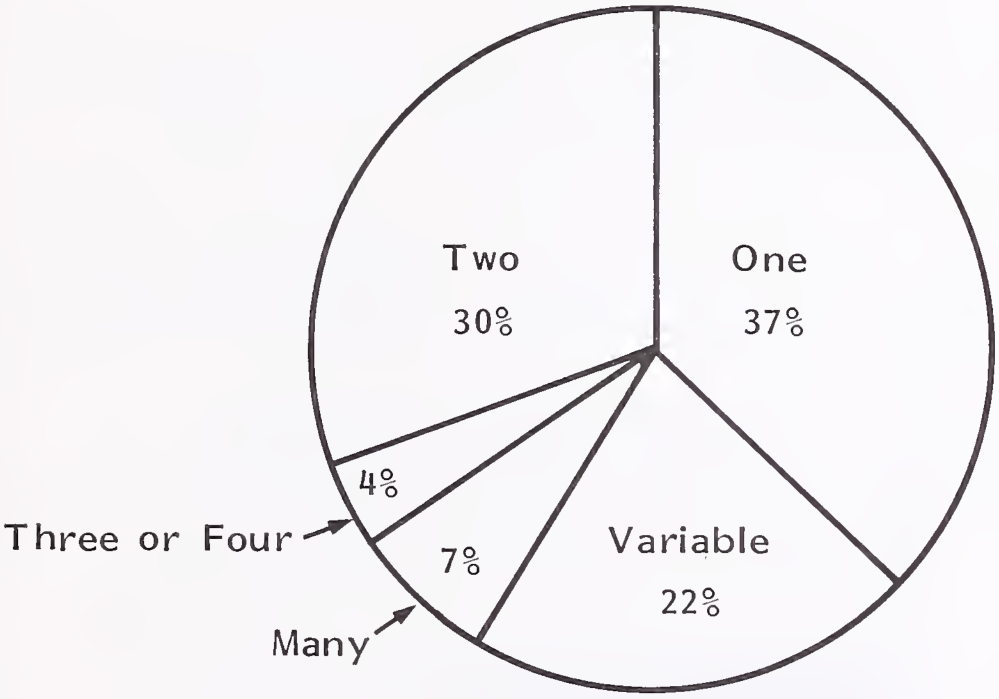
  . Downloading software revisions.

- Hardware vendors are five times as likely to do so.

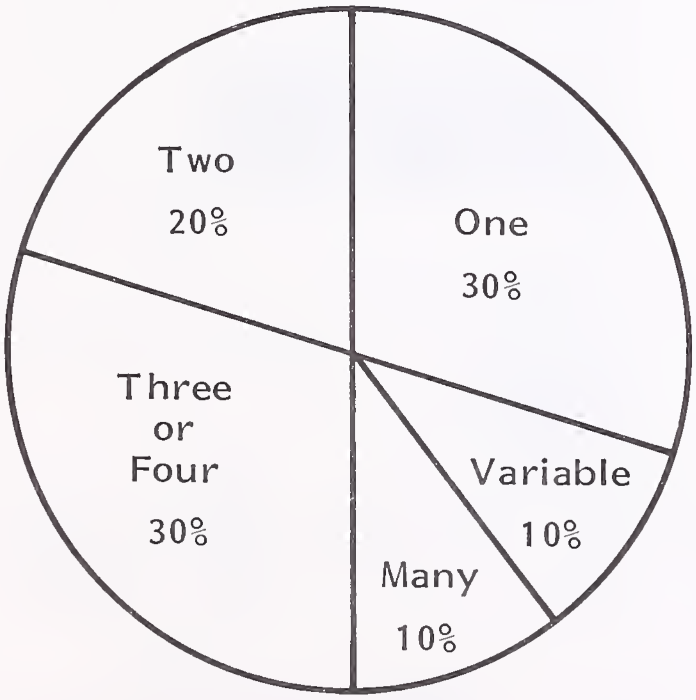
- About half the software vendors have plans to use telecommunications for these purposes eventually, while at least 80% of the hardware vendors plan to do so eventually.

EXHIBIT III-12

NUMBER OF SOFTWARE REVISIONS DISTRIBUTED ANNUALLY



SOFTWARE VENDORS

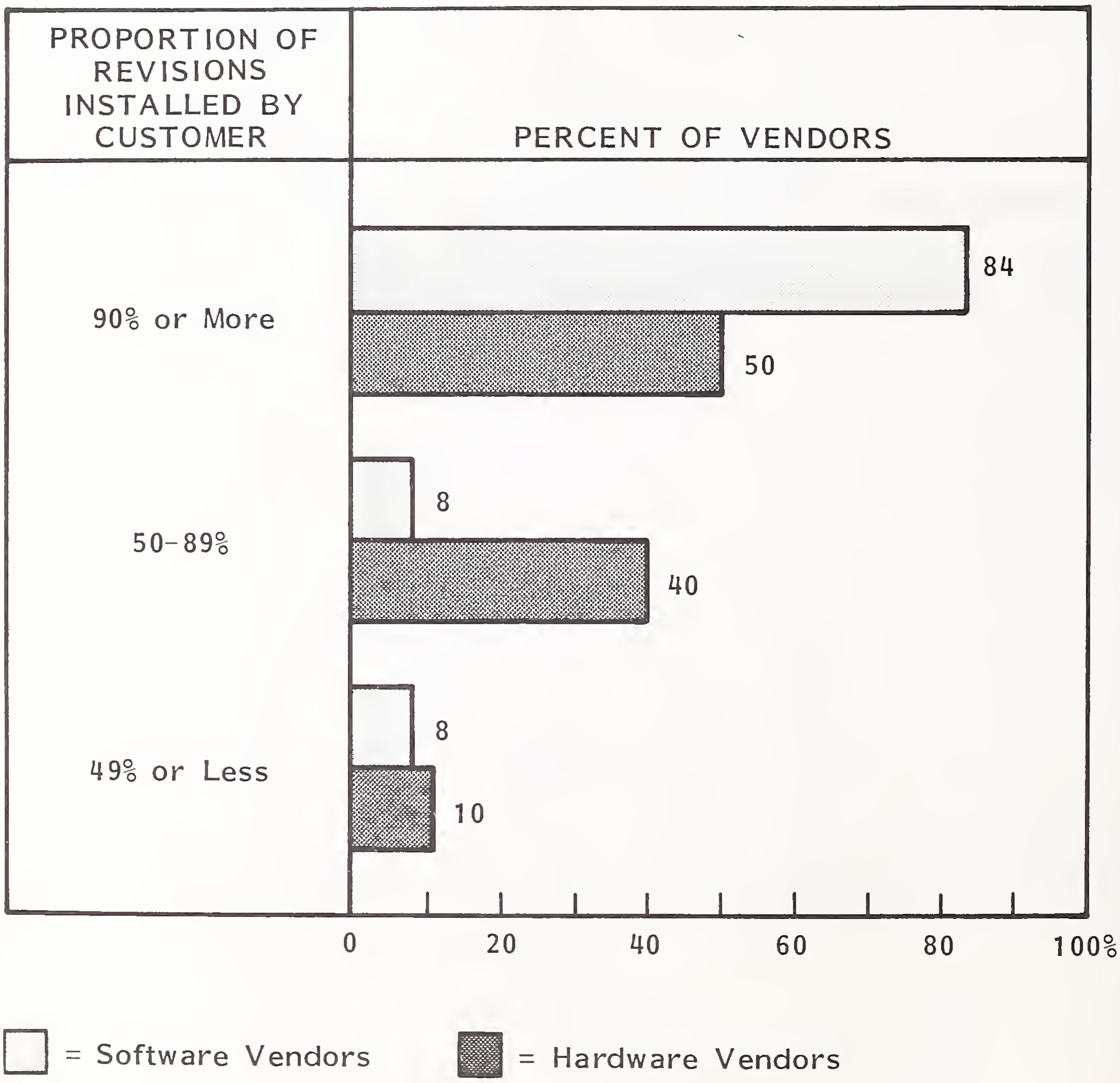


HARDWARE VENDORS

SOURCE: INPUT Survey

EXHIBIT III-13

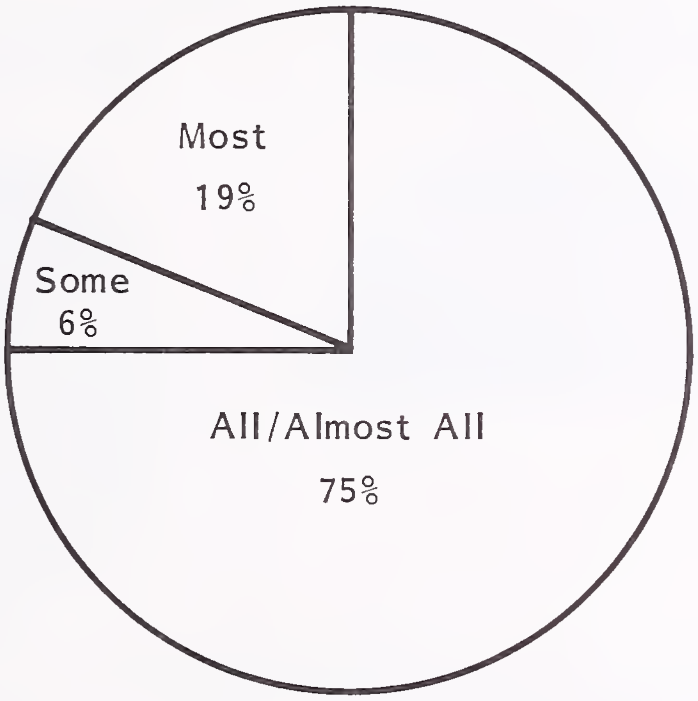
SOFTWARE REVISIONS INSTALLED BY CUSTOMER



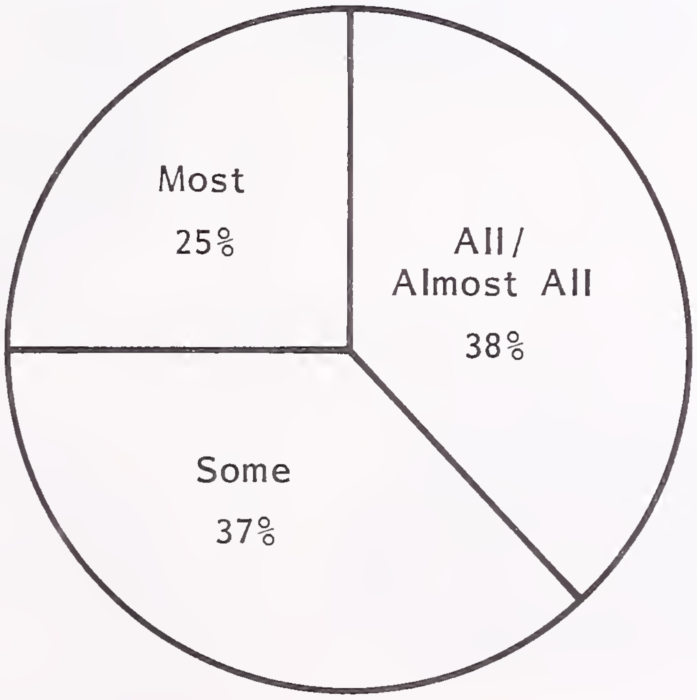
SOURCE: INPUT Survey

EXHIBIT III-14

PROBLEMS RESOLVED BY REGULAR SOFTWARE REVISION CYCLE



SOFTWARE VENDORS

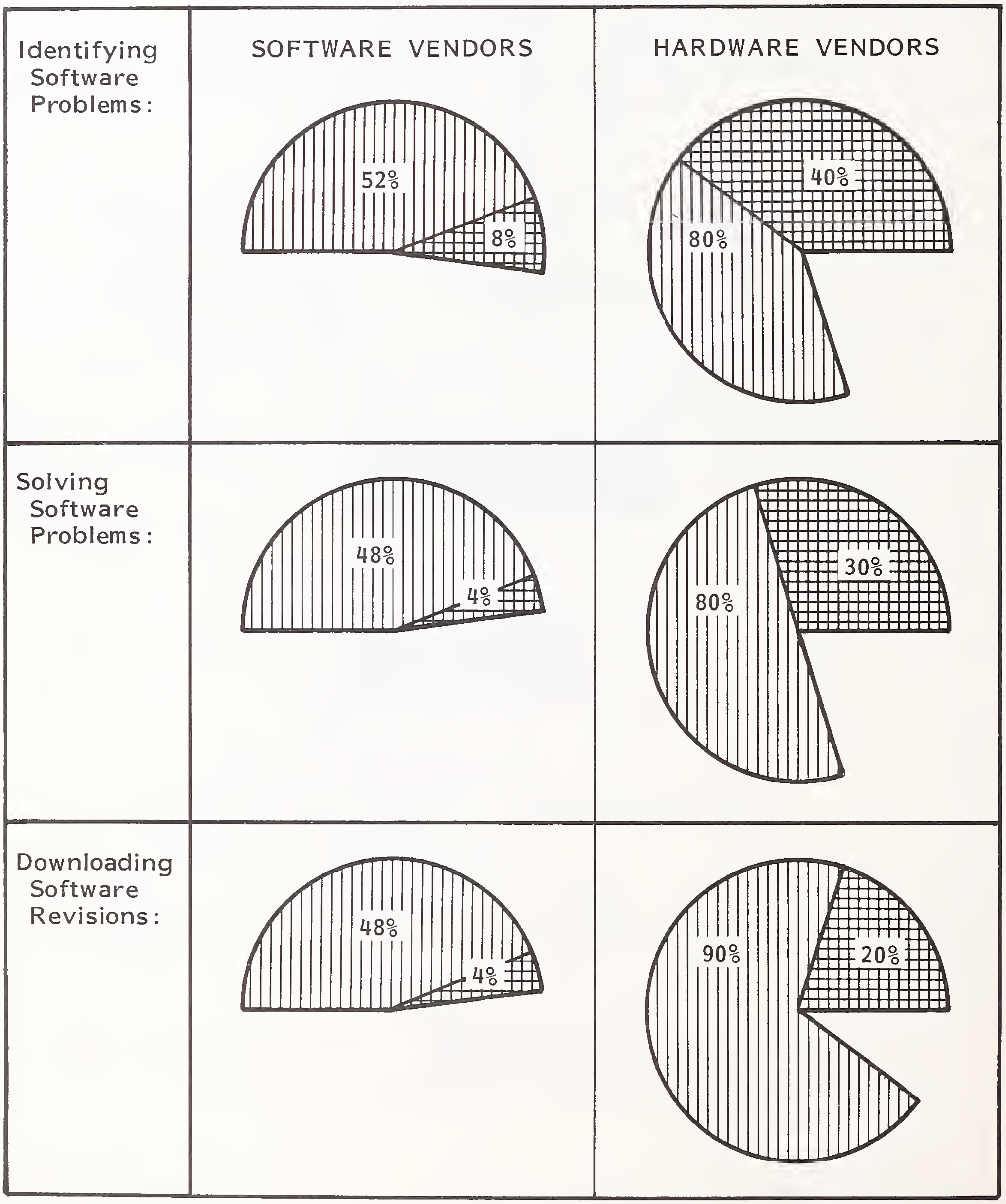


HARDWARE VENDORS

SOURCE: INPUT Survey

EXHIBIT III-15

PERCENT OF VENDORS USING AND PLANNING TO USE TELECOMMUNICATIONS IN SOFTWARE MAINTENANCE



▤ = Now

▥ = Future

Note: Percents refer to vendor "always" or "usually" using or planning to use

SOURCE: INPUT Survey

IV SOFTWARE MAINTENANCE ORGANIZATION AND STRATEGY

IV SOFTWARE MAINTENANCE ORGANIZATION AND STRATEGY

A. ORGANIZATIONAL ISSUES

1. THE CUSTOMER SERVICE FUNCTION

- In virtually all companies the software customer support staff is ultimately attached to the marketing organization.

  - The typical hardware company organizes its software field support organization as shown in Exhibit IV-1. A few companies, such as Honeywell, have recently transferred software maintenance responsibilities to field engineering; i.e., the hardware maintenance organization.

  - A number of other hardware companies have been debating the value of a similar transfer.

    - INPUT's observation is that such transfers are neutral. There are advantages and disadvantages in having the software maintenance in marketing (which in effect means that it is semi-independent) as well as having it in field engineering. Exhibit IV-2 shows some of the pros and cons.

    - The value of such a transfer will depend largely on the status of a company's marketing and field engineering organizations at a particular time, and the attitudes of key personnel.

EXHIBIT IV-1

SOFTWARE FIELD SUPPORT ORGANIZATION IN A TYPICAL HARDWARE COMPANY

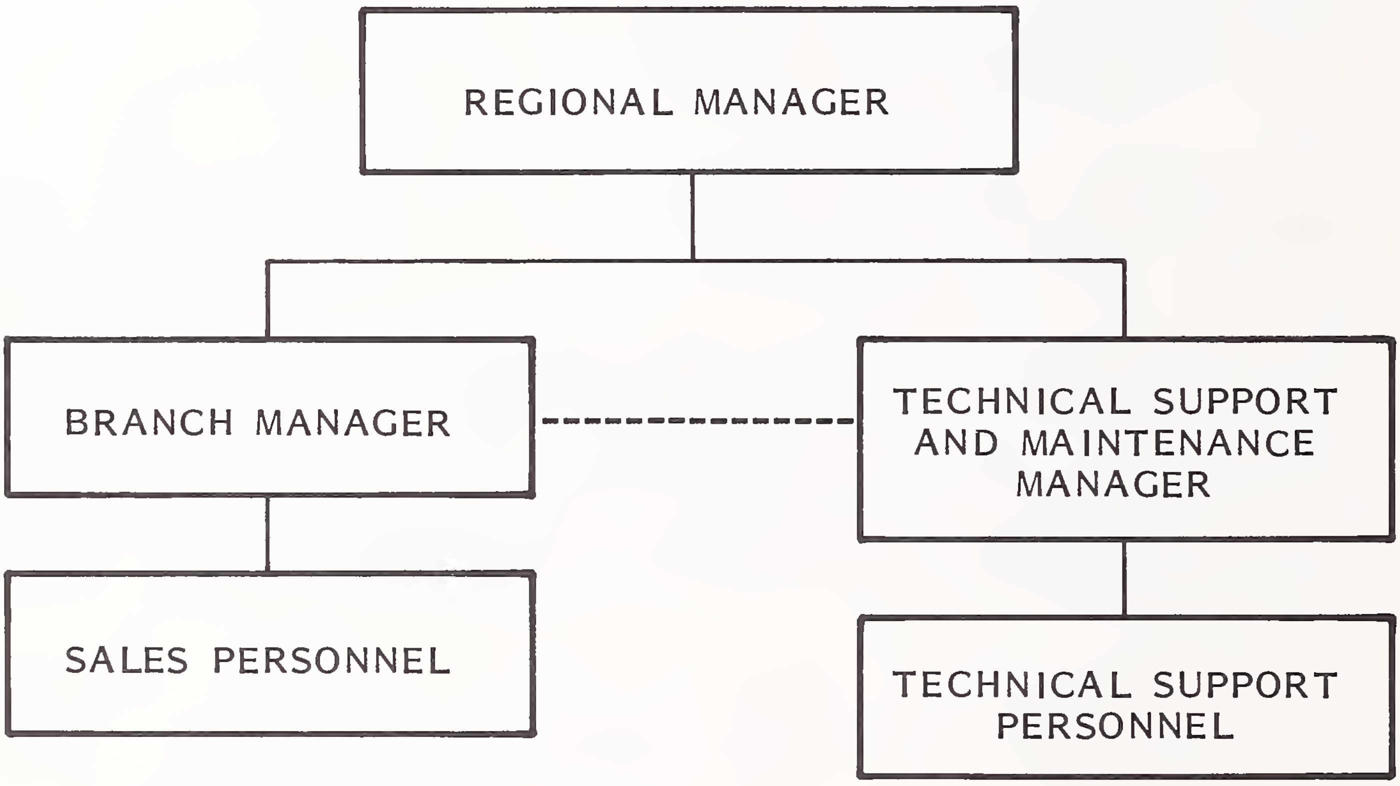

REGIONAL MANAGER
BRANCH MANAGER
TECHNICAL SUPPORT AND MAINTENANCE MANAGER
SALES PERSONNEL
TECHNICAL SUPPORT PERSONNEL

EXHIBIT IV-2

ORGANIZATIONAL LOCATION OF SOFTWARE MAINTENANCE CUSTOMER SERVICE FUNCTIONS

	MARKETING	FIELD SERVICE
Advantages	Maintenance is integrated with pre- & post-sales support Maintenance activities can directly support sales efforts Marketing can understand customer product needs better	All maintenance activities are co-located Staff can be cross-trained Hardware maintenance staff can be retrained for software
Disadvantages	Marketing is not technically oriented Potential conflict between sales support and maintenance Marketing management may emphasize sales activities	Hardware and software technical issues is much different Inter-departmental cooperation needed to sales support Hardware retraining is difficult

- It is important to keep in mind that in certain critical respects software maintenance does not fit well in either marketing or field engineering (at least as it is presently constituted).

    - Software in general is unlike hardware, as shown in Appendix D).

    - Software maintenance will always have ties to the software development function.

    - People in software have different personal characteristics and attitudes from people in marketing and field engineering.

- Regardless of the organizational sponsorship, communication between the customer and the central maintenance group will follow the process shown in Exhibit IV-3.

    - The customer support representatives are not necessarily technically trained in the internals of the product, but have an excellent hands-on knowledge of the product from the user's perspective.

        - If the vendor has a large enough customer base and resources, the representatives will specialize by product.

        - The staff can also provide sales and installation support.

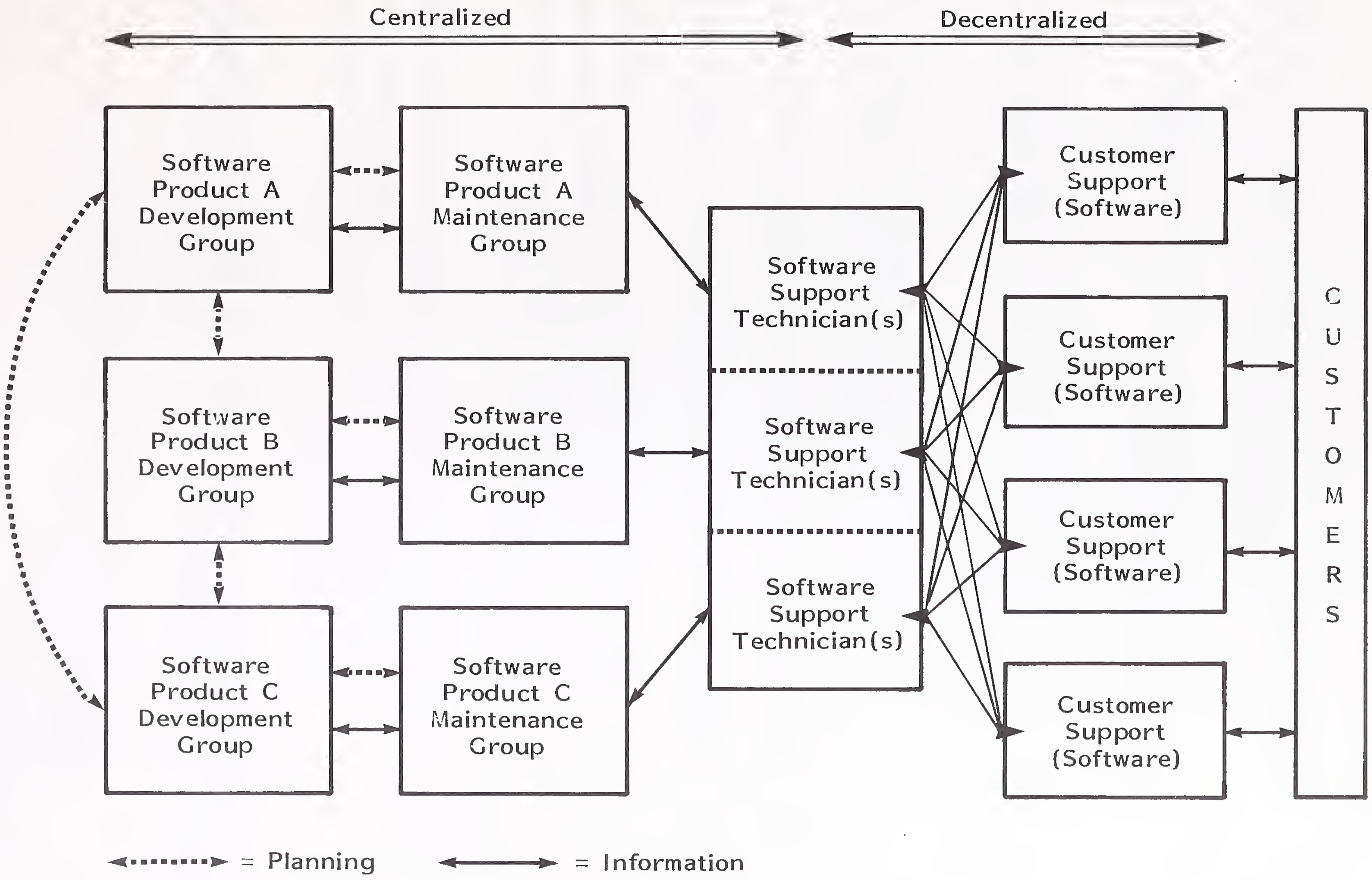
        - Personality is more important than intellectual skills.

    - The software support technicians are "middlemen".

        - They back up the customer supports reps on narrow or technical issues.

EXHIBIT IV-3

SOFTWARE MAINTENANCE COMMUNICATIONS


Centralized
Decentralized
Software Product A Development Group
Software Product A Maintenance Group
Software Product B Development Group
Software Product B Maintenance Group
Software Product C Development Group
Software Product C Maintenance Group
Software Support Technician(s)
Software Support Technician(s)
Software Support Technician(s)
Customer Support (Software)
Customer Support (Software)
Customer Support (Software)
Customer Support (Software)
CUSTOMERS
= Planning
= Information

- They must specialize by product.

- They are filters to the central maintenance group.

- The maintenance group is made up of true software technicians (programmers and analysts).

  - They must keep up some contact with the field staff (and even customers) so that they do not become divorced from the real world.

  - They, in turn, must interact with the new product development group. This relationship is the subject of the next section.

2. THE RELATION OF THE DEVELOPMENT AND MAINTENANCE FUNCTIONS

- The role of the software support staff varies from company to company.

  - Generally speaking, the support staff is highly (usually, solely) involved with customer contact, as well as error determination and correction. There is less involvement in the design coding and documentation of software revisions, as shown in Exhibit IV-4.

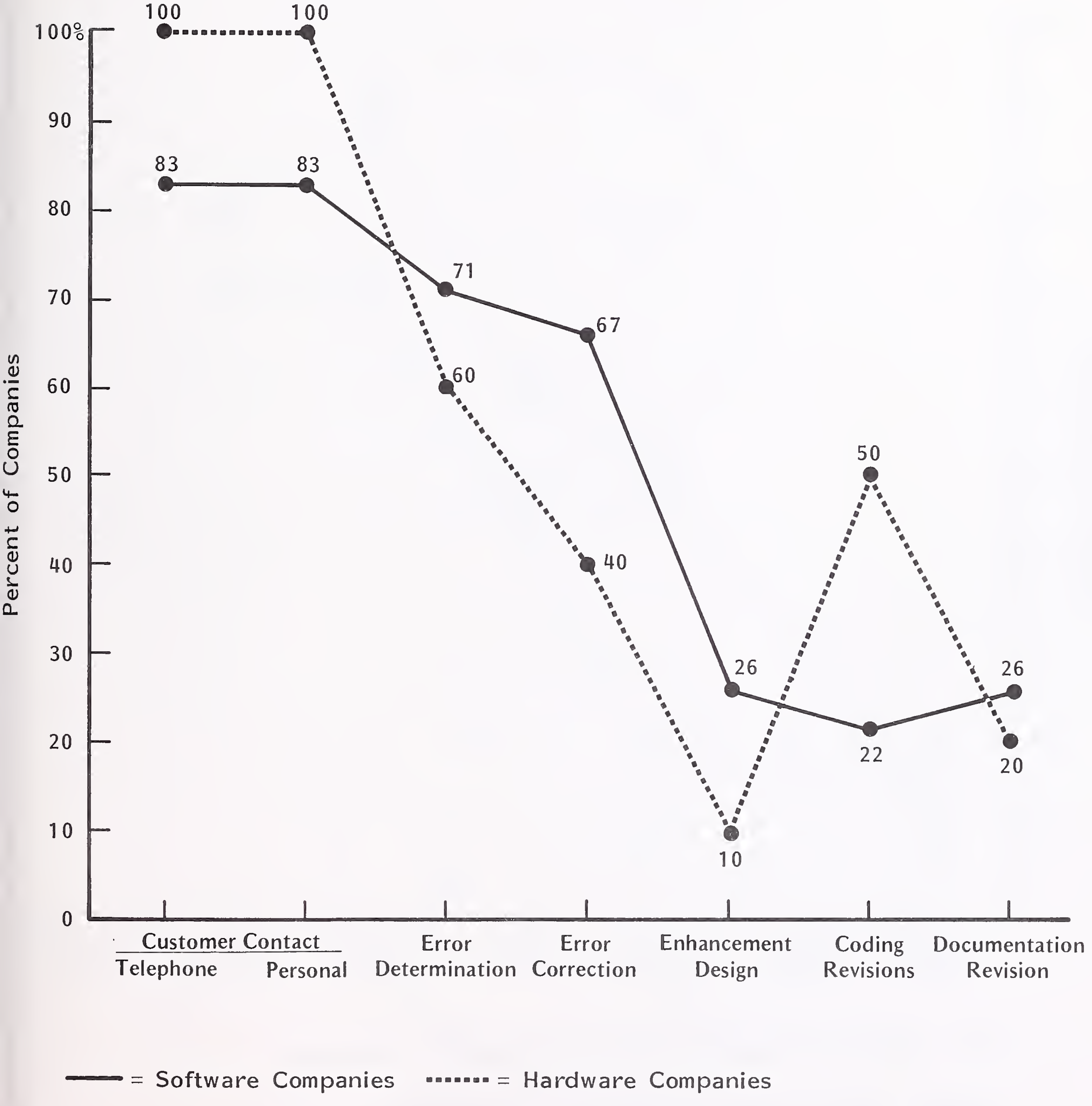
  - The software development group mirrors the support group's involvement, as shown in Exhibit IV-5. Development groups in hardware companies tend to be more involved than those in software companies, as shown in Exhibit IV-6.

  - Respondents express satisfaction with present arrangements and plan few changes.

- There are three basic models for organizing the central software maintenance function:

EXHIBIT IV-4

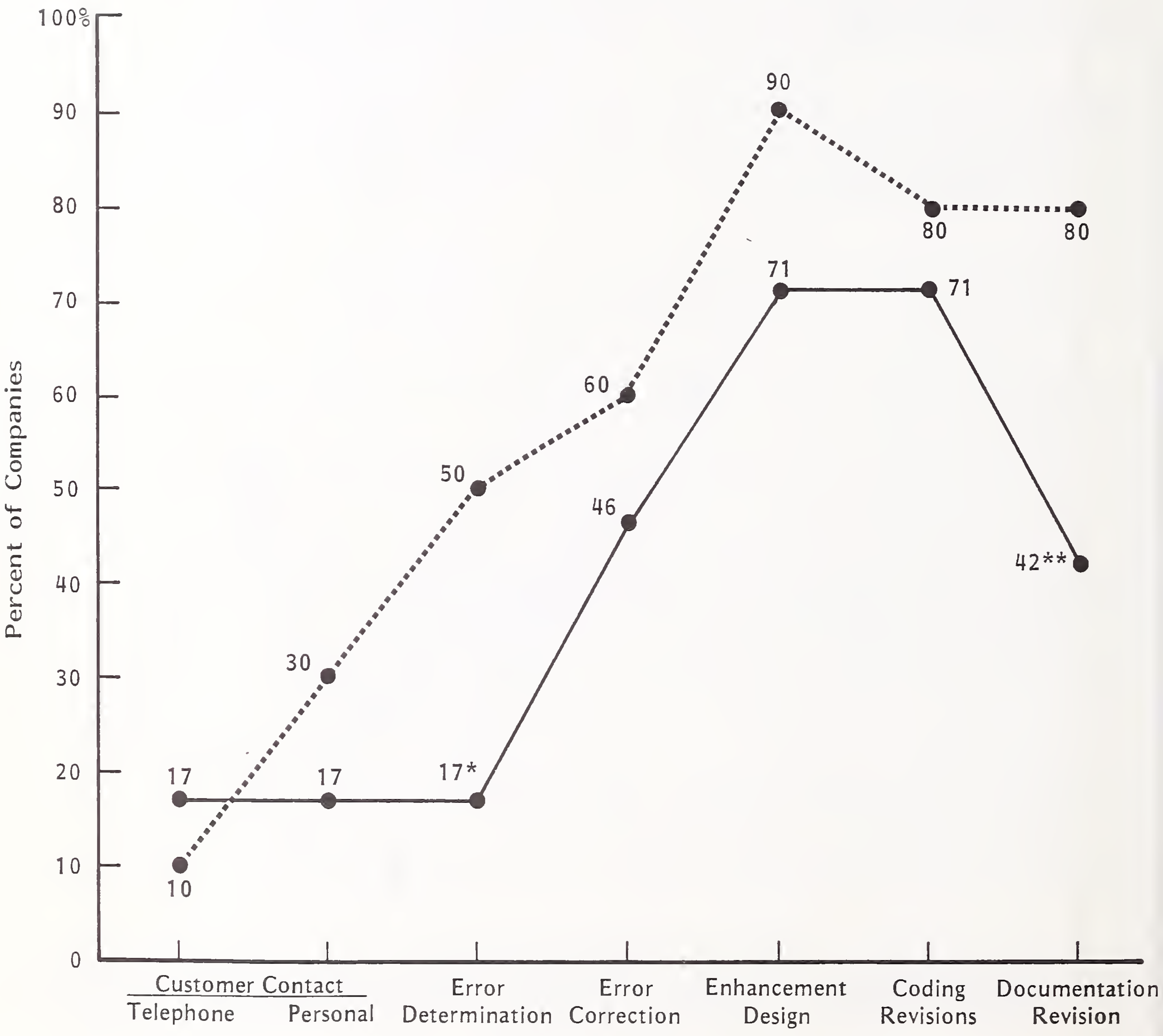
SOFTWARE SUPPORT STAFF FUNCTIONAL INVOLVEMENT



SOURCE: INPUT Survey

EXHIBIT IV-5

SOFTWARE DEVELOPMENT GROUP FUNCTIONAL INVOLVEMENT



—— = Software Companies ········ = Hardware Companies

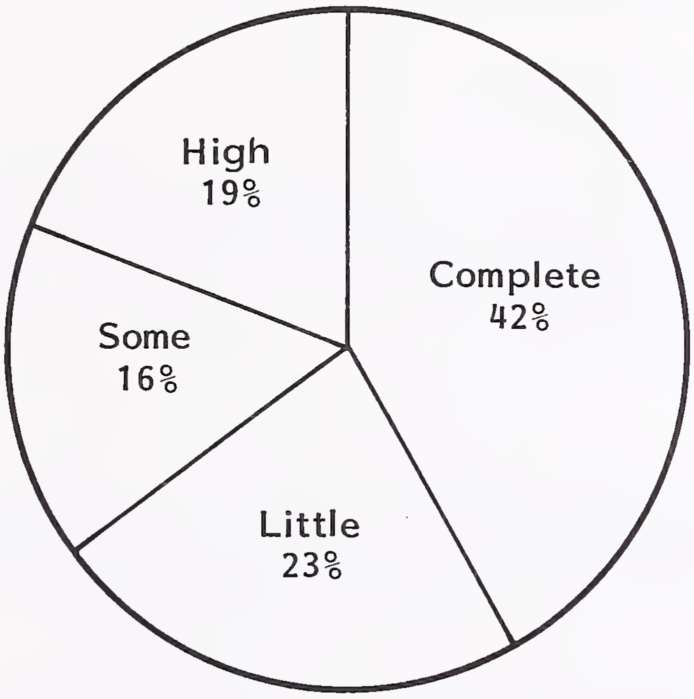
* Quality Control also involved in 17% of companies

** Documentation Group also involved in 21% of companies

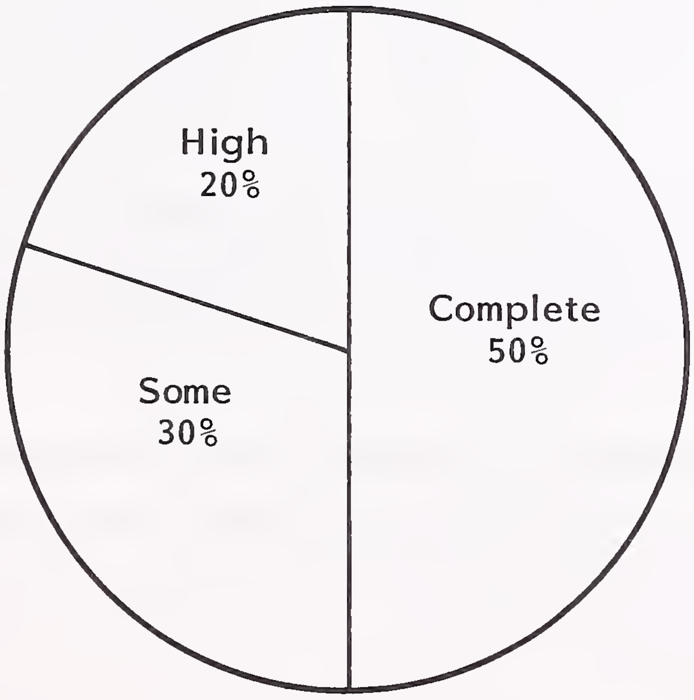
SOURCE: INPUT Survey

EXHIBIT IV-6

EXTENT OF INVOLVEMENT OF DEVELOPMENT GROUP IN MAINTENANCE



SOFTWARE VENDORS



HARDWARE VENDORS

SOURCE: INPUT Survey

- These approaches can be labeled as:

  - Coordinated.

  - Integrated.

  - Independent.

- In the coordinated approach software maintenance and development are separate entities, but they report to the same product software manager.

- The integrated approach is similar, except that the developer and maintainer roles are not distinct. There is much trading of responsibilities. No one is stuck doing maintenance.

- The independent approach separates developers and maintainers. Separate maintenance career paths and specializations can be developed. This is not practical if the entire staff for a software product (or product group) is small; i.e., under 25.

- Exhibit IV-7 shows the three approaches graphically.

- Each organization option has different effects on the turnover, morale, skills, and feasible project size of the central software maintenance organization. Exhibit IV-8 summarizes these effects.

  - The independent organization is the most conducive to effective maintenance activities. However, skills are needed to coordinate software activities across a product. The development group will usually oppose this approach.

EXHIBIT IV-7

ORGANIZATIONAL ALTERNATIVES FOR CENTRAL SOFTWARE MAINTENANCE FUNCTION

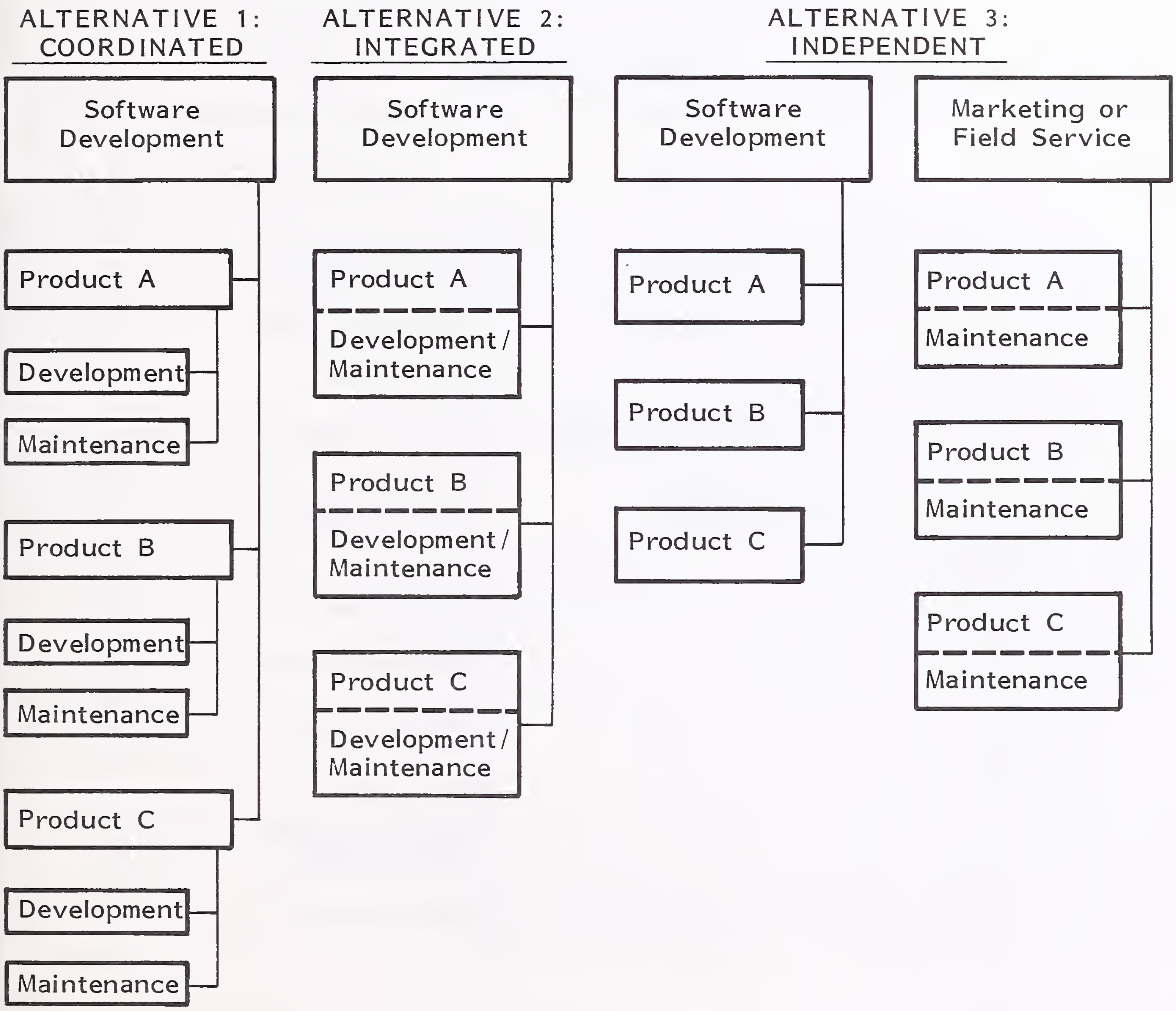


EXHIBIT IV-8

EFFECTS OF SOFTWARE MAINTENANCE ORGANIZATION OPTIONS

		MAINTENANCE PHYSICALLY AND ORGANIZATIONALLY DISTINCT	
		Yes	No
MAINTENANCE ORGANIZATION PERFORMS MAINTENANCE ONLY	Yes	(Independent) Low Turnover High Morale High Skills Developed Large Project Size Feasible	(Coordinated) High Turnover Low Morale Medium Skills Developed Large Project Size Feasible
	No	N/A	(Integrated) Medium Turnover Medium Morale Medium Skills Medium Project Size Feasible

- The integrated approach is well-suited to small software groups. The problem is that no one wants to do maintenance, and the integrated approach often degenerates into the coordinated approach.

B. STRATEGIC ISSUES

1. ASSESSING STRENGTHS AND WEAKNESSES

- In developing a strategy for approaching software maintenance, the first questions to ask are: "What kind of company am I?" then, "Will I be the same company in five years?"

  - The obvious place to start is with the differences between hardware and software companies. Some of the advantages and disadvantages of each are summarized in Exhibit IV-9.

    - Many of the strengths and weaknesses of software companies are due to their relatively small size.

    - Hardware companies, especially the mainframe companies, are more ponderous and structured organizations. This is not always a disadvantage for a support function. Customers expect support to be uniform and by-the-numbers. There is little room for inspiration in a support environment.

  - Exhibit IV-9 should not be taken as a prescription for every company. Each company is in a unique situation. Ideally, each company will make its own list of strengths and weaknesses and look for ways to build on its strengths and, at the least, minimize its weaknesses.

EXHIBIT IV-9

SOFTWARE MAINTENANCE STRENGTHS:

HARDWARE COMPANIES AND INDEPENDENT SOFTWARE COMPANIES

	INDEPENDENT SOFTWARE COMPANY	HARDWARE COMPANY
Advantages	High profile in area of specialization Deep knowledge of products and market in area of specialization Quick reactions to market Relatively easy for new entrants to produce products Attractive to entrepeneurial/ risk-taking staff attempting breakthrough products	Large resource base (dollars, people) Integrated, comprehensive software products Good geographic support for marketing and support Sole source for some systems software Much closer integration of hardware and software
Disadvantages	Limited resource base (dollars, people) Product line usually limited Difficult to obtain satisfactory marketing and support geographic coverage Relatively easy for new entrants to produce products Must react to hardware changes	Resources possibly spread too thin Software products possibly obsolescent, inadequate, or nonexistent Reaction time may be slow Risk taking may not be welcomed Software traditionally only offered for own hardware

2. MAINTAINING THIRD PARTY-DEVELOPED SOFTWARE

- In the past, vendors tended to develop their own software. There was then little question, or option, of who would maintain the software.

- This situation is now changing as more companies are adding specific software products from outside suppliers to their own line of products.

  - One alternative, followed by many minicomputer and small system vendors, is not to actually acquire the software, but to keep at arm's length from the vendor.

    - At the most, the hardware vendor examines the software and recommends its use.

    - At the least, the hardware vendor merely maintains lists of software products but makes no recommendations of one over another.

    - Either way, the hardware vendor has little control over the product's evolution, its quality, or even its existence.

  - The alternative, followed by such diverse companies as IBM and Cullinane, is to buy up rights to a product. Where product presence is desirable, this gives a vendor a proprietary product and complete control over it.

- The question then becomes: will the buying or selling company maintain the software?

  - The main reason for going outside in the first place is to lower the investment in time and resources to develop a software product.

- Contracting with the seller to continue supplying central maintenance functions would lower initial investment.

- It may be possible as part of the acquisition to take on part of the seller's technical and maintenance staff. This is a desirable alternative, if feasible. However, many people will not want to leave their company or will not last long in a new, usually much larger company.

- Exhibit IV-10 summarizes the pros and cons of having the buyer or seller maintain third party-developed software.

3. NEW VERSUS ENHANCED SOFTWARE PRODUCTS

- One of the barriers to making software maintenance into a functioning P&L center is that some of the most attractive enhancements to existing software can just as easily be packaged as new products. If this is done, the benefits do not accrue to the software maintenance organization.

- Many software planners freely admit that their firms do not have hard and fast rules for deciding when a bundle of capabilities is a new product as opposed to an enhancement, or what constitutes a major as opposed to a minor enhancement. Exhibit IV-11 shows the relationship, and overlap, between new product development and maintenance enhancements.

- Existing customers, of course, want all possible product additions to be considered enhancements and included as standard revisions covered by their maintenance contracts. Older customers (and some old-time vendor personnel) identify with the bundled software era when everything was "free".

  - In reality, customers have little or no contracted protection from vendors announcing an "improved" software product, and charge current customers a significant proportion of list price, if not the full list price.

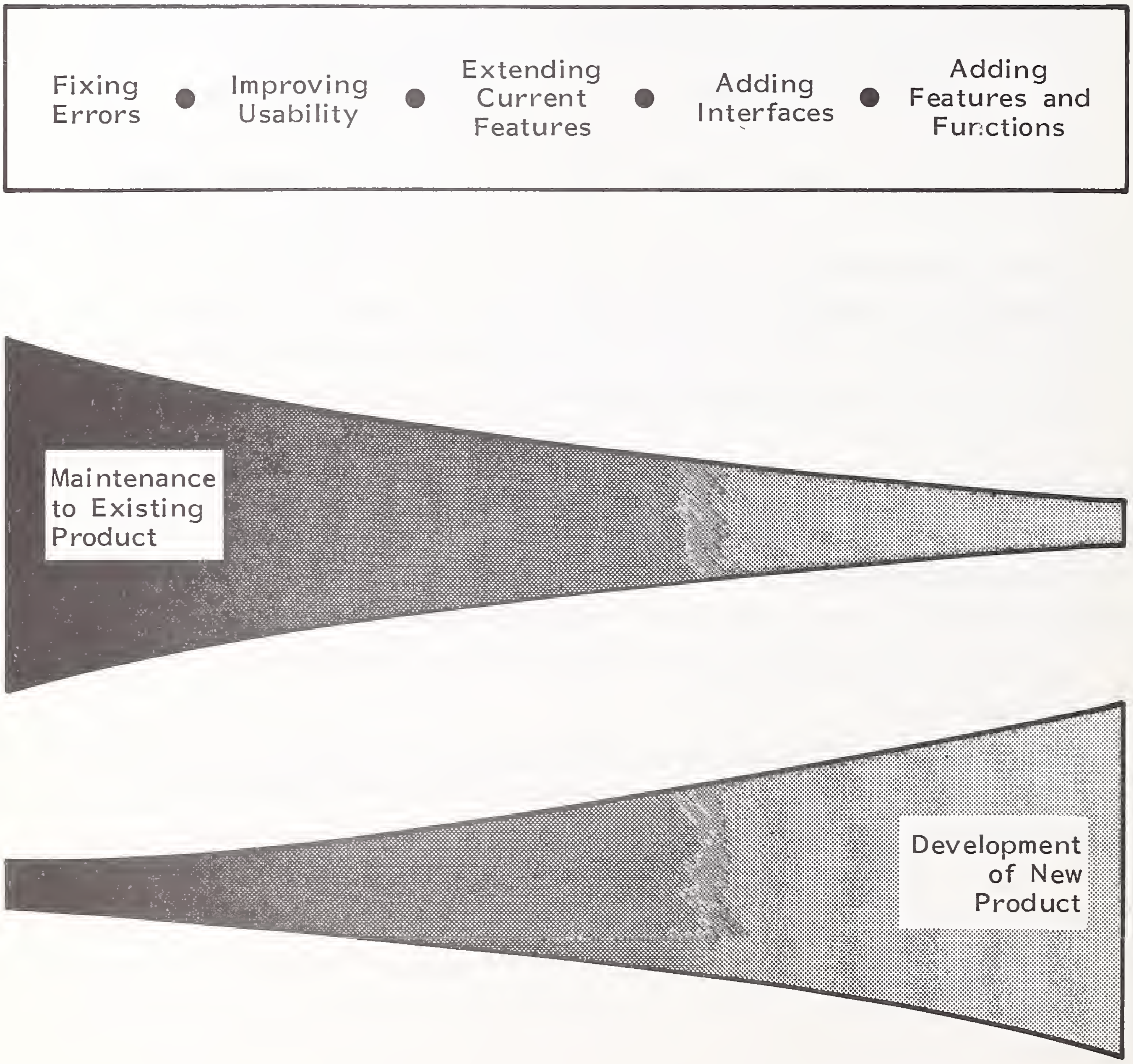
EXHIBIT IV-10

MAINTAINING THIRD PARTY-DEVELOPED SOFTWARE: BUYER OR SELLER?

ADVANTAGES TO BUYER MAINTAINING	ADVANTAGES TO SELLER MAINTAINING
● More direct quality control ● Easier maintenance of documentation and other standards ● Possible addition of key seller staff ● Difficulty in motivating staff for maintenance ● Easier field-central staff coordination	● Lower initial investment in resources and management time ● Conserve scarce in-house staff ● Greater expertise of seller's staff ● Reluctance of seller's staff to join/stay with buyer

EXHIBIT IV-11

DEGREE OF INVOLVEMENT OF NEW PRODUCT DEVELOPMENT AND MAINTENANCE


Fixing Errors
●
Improving Usability
●
Extending Current Features
●
Adding Interfaces
●
Adding Features and Functions
Maintenance to Existing Product
Development of New Product

- The only barrier to this (but it is a strong one) is the long-term damage it will do to the vendor's standing in the marketplace. Some vendors have damaged their reputations in this way, usually because of serious financial pressures.

- Some vendors adopt a middle path, announcing a higher-priced, improved product, while including many of the new features as maintenance revisions to current products.

  - This approach must be thought out well from a marketing standpoint so that satisfying current customers does not undermine future sales.

  - There is a long-term technical burden in maintaining two or more similar, but not identical, products.

- For this reason many vendors "bite the bullet" and make it financially advantageous to upgrade to new products, especially when the old product, at a technical dead end, will not attract many new sales in any event.

  - Negative incentives can also be applied by announcing that support of the old product will be stopped soon (generally in less than a year).

  - This will get the new product off to a rousing start by giving it an instant track record.

4. ASSESSING SOFTWARE MAINTENANCE OPPORTUNITIES

- Not all software packages are created equal, from a software maintenance standpoint.

  - Few customers will want to go "bare" on operating system maintenance, even if they have the chance.

- On the other hand, many purchasers of large, industry-specialized packages buy the package intending to modify it extensively. For them, maintenance is just a tax on the purchase price.

- A buyer of small, stable packages that have been in existence for some time will rarely feel the need for extensive maintenance.

- Maintenance is perceived as highly valuable in large, complex packages that the customer has no intention of modifying. DBMS is a good example of this type of product.

- These relationships can be graphically illustrated, as shown in Exhibit IV-12.

- This is not to say that vendors should ignore the low-need areas. These can in fact be the most profitable (at least in the near term, as shown in Chapter V). Two approaches can be made:

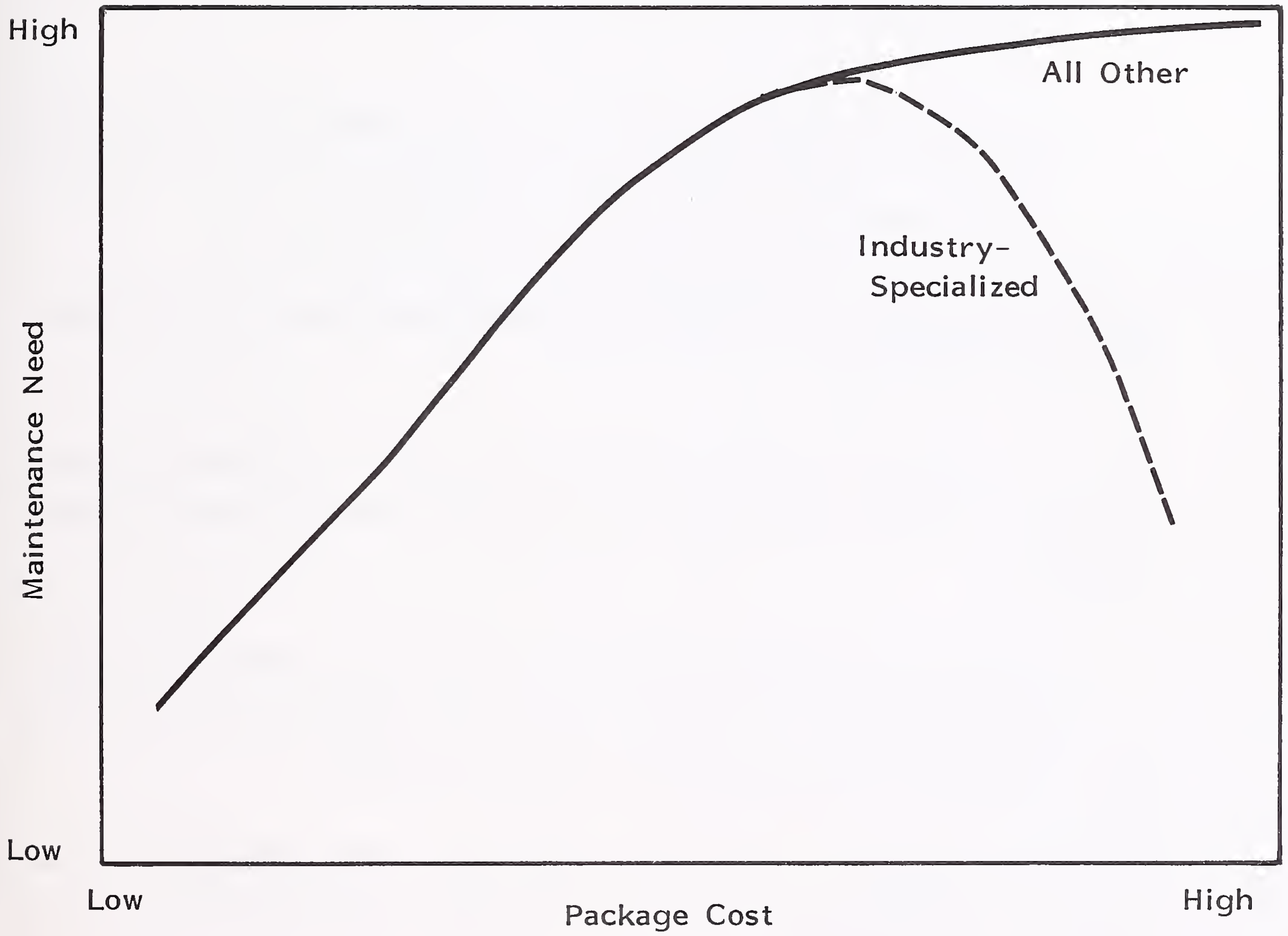
  - Tax: Given the relative price-insensitivity to software, if a customer sees a need for a package at $X, then the customer will usually not balk at an additional $.1X per year. If the vendor has an attractive product, then there should be a mandatory maintenance requirement, at least for several years.

  - Insurance Policy: The other approach, useful for small stable packages, is to have a nominal maintenance price, covering error fixes only. At the right price, customers will buy the insurance for at least several years.

- Vendors should keep in mind that historically maintenance in general has not been tracked or controlled very precisely in many data processing budgets.

EXHIBIT IV-12

SOFTWARE MAINTENANCE NEEDS



- In some companies, operations software (and its maintenance) is included in hardware budgets, a remnant of bundled hardware days.

- In other companies, hardware and software maintenance budgets are combined.

- In still others, specific application software and maintenance expenses are charged to a particular application system.

- In many companies, these different classifications are being used simultaneously.

- There have been some attempts to tighten up as a result of the current recession; however, software maintenance is too scattered and misunderstood for cost cutting to have much effect.

5. THE IMPORTANCE OF SOFTWARE MAINTENANCE IN A COMPANY'S BUSINESS STRATEGY

- Software maintenance can be an important part of a company's business strategy. Software maintenance is in some ways the last frontier for many companies.

  - Hardware's price/performance and reliability are improving rapidly. Unfortunately for established vendors, these same factors are turning many hardware products into a commodity market.

    - Hardware maintenance has been most resistant to these tendencies, but even here third-party maintainers have gotten a foothold.

    - Over the longer term, rapidly falling prices and increasing reliability will reduce hardware maintenance opportunities

- Software in general is a messy area. It is expensive to produce and often only meets customer needs marginally. Software productivity has lagged far behind hardware performance. New software products, especially for smaller systems, are easy to produce in the well-known garage.

  - This places pressure on established vendors.

  - Outsiders cannot, however, feasibly offer add-ons or maintenance to existing software products. Consequently, it is the most protected area for established vendors.

- Exhibit IV-13 summarizes these relationships and trends.

- The conclusion is that software and especially software maintenance are tough areas for developing satisfactory products economically. Vendors who can make even marginal breakthroughs should be able to reap large rewards.

EXHIBIT IV-13

STRATEGIC FACTORS FOR HARDWARE AND SOFTWARE PRODUCTS

	COST TO PROVIDE	CUSTOMER NEEDS SATISFACTION	RELIABILITY	RESISTANCE TO NEW COMPETITORS
Hardware Products	++	++	+	—
Hardware Enhancements	++	++	+	—
Hardware Support	+	0	+	+
Software Products	—	—	0	—
Software Enhancements	—	—	0	++
Software Support	— —	— —	—	++

Key: ++ = Very Favorable
+ = Favorable
0 = Neutral
— = Unfavorable
— — = Very Unfavorable

V PLANNING ISSUES

V PLANNING ISSUES

A. TRENDS

- Trends already under way could have a significant impact on software maintenance planning. These are coming from opposite ends of the computing spectrum; i.e.:

  - The impact of personal computer software.

  - The breakup of the IBM system software monopoly in the plug-compatible market.

1. PERSONAL COMPUTER SOFTWARE

- The evolving pricing structure of personal computer/small computer software is far different from that for mainframes, or even minicomputers. To take financial planning software as an example:

  - Mainframe-based packages (e.g., EXPRESS, EMPIRE, MAPS) range in price from $20,000 to $200,000.

  - Personal computer-based packages (Visi- and other "calcs") are two or three orders of magnitude cheaper. A single VisiCalc package can be purchased for about the cost of one month's maintenance for some of the mainframe-based packages; it is easy to see why the product manager for one package sees VisiCalc as his major long-term competition.

- Given these prices, neither customers nor vendors of most personal computer software expect much in the way of traditional maintenance:

  - Customer expect to live with minor bugs and other difficulties.

  - At best, previous purchasers can look forward to a discount on subsequent revisions to the product (which may be nothing more than the current product with bugs removed).

  - Often, personal computer software is delivered without even a telephone number or address for followup contact. The dealer is expected to provide installation and followup support; obviously, this support is often minimal.

- Obviously, a totally "let-the-buyer-beware" environment cannot persist. However, personal computer application software will habituate a new and very large generation of users to a minimal level of maintenance, compared to traditional software.

- An equally important change is the lack of coupling among hardware, operating system, and application software.

  - In a mainframe-based system, application software must typically be tailored for a particular hardware environment. Most software is aimed at a limited number of environments, often just the IBM mainframe environment.

  - The original personal computer entrants (Apple, Tandy) preserved this captive applications software approach: Visicalc, for example, originally only ran on Apple. However, this is beginning to fade.

- The IBM personal computer, for example, was announced with three operating systems (all from independent suppliers).

- The contrast between these two approaches is shown in Exhibit V-1.

- The falling price of hardware has pushed this trend to its logical extreme, with new models (e.g.; Franklin ACE) or add-on boards available to mimic the operation of other manufacturers' hardware environments.

- The lesson from the personal computer marketplace is that soon little software will be automatically tied to a particular manufacturer: operating systems will be in the same category as application programs.

- The loosening of operating system monopolies is most evident in personal computers. However, the popularity and spread of UNIX, UNIX-like systems (Xenix, UNOS, etc.) and the PICK operating system show similar trends in the minicomputers.

2. IBM MAINFRAME OPERATING SYSTEMS

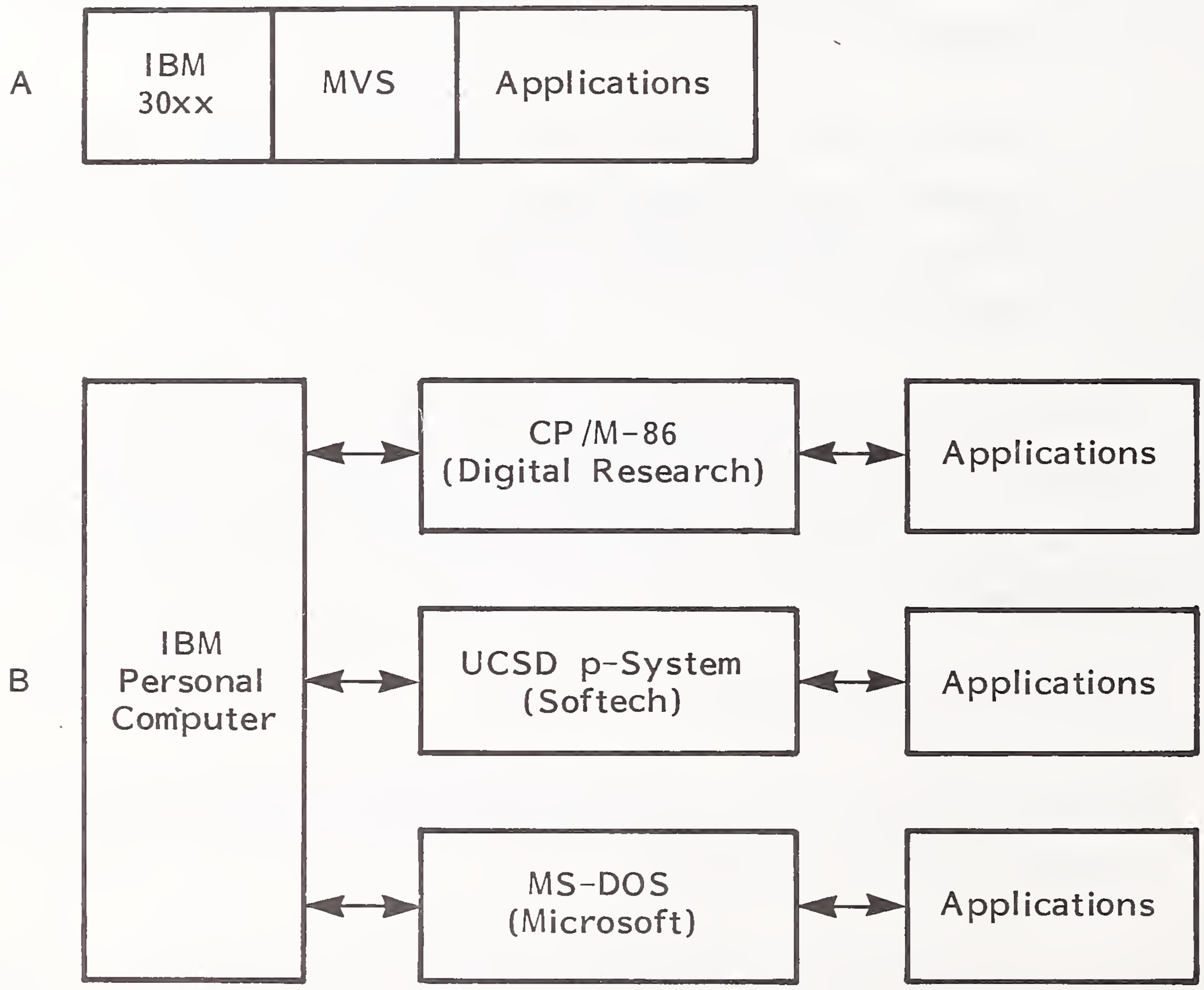
- For the first time in the computer era there are signs of change in the way operating systems are supplied.

- Heretofore, each mainframe manufacturer (IBM plus the "BUNCH") supplied their own operating system(s) that were incompatible with those of other manufacturers.

- The advent of the plug-compatible manufacturers did not change this appreciably, since IBM-supplied operating systems would still run on all systems.

EXHIBIT V-1

CONTRASTS IN HARDWARE/SOFTWARE LINKAGES
(Using IBM as an Example)



- Competitive pressures are beginning to change this. IBM has already pulled some functions back into microcode. More importantly, its competitors fear that additional, unannounced functions are already in place in microcode.

  - This has led Amdahl to develop, in spite of some current difficulty, its own implementations of microcode/operating systems.

  - The Japanese have for some time been producing mainframes containing some non-IBM instructions; their computers have emulated IBM's. Now there are indications that the Japanese will also be breaking away from one-for-one interchangeability with IBM operating systems.

- These developments may lead to the interesting situation of perhaps half a dozen IBM-derived operating systems, with significant overlap and commonality. Presumably, these variant operating systems will be transparent to most or all current IBM-oriented applications and operating support software.

  - This should create significant competition in what could be termed the "plug-modifiable operating system" market.

  - Operating system users would no longer be captives. Where warranted, users could switch operating systems in the same way that plug-compatible hardware is changed now.

  - Operating system maintenance would be forced to emerge from the technical area to the marketing front lines.

B. IDENTIFYING AND ADDING VALUE TO SOFTWARE MAINTENANCE

- Pricing will be the key continuing issue in software maintenance.

- Vendor costs, perceived value to the customer, and customer price insensitivity act together to push prices up.

- Competition and customer price sensitivity act to push prices down.

- Exhibit V-2 illustrates how these forces act on both the price floor and price ceiling.

- The trends and competitive forces discussed in the previous section of this chapter will act together to sensitize customers to software maintenance pricing.

- This will place a heavier burden on perceived value than previously. Customers will begin to evaluate exactly what they are receiving as "software maintenance".

- One approach, which should be useful to vendors and customers alike, is to break software maintenance down into its constituents. While the constituents may vary from product to product, the following categories will serve most analyses:

  - Error correction/prevention.

  - Improvements to featurers.

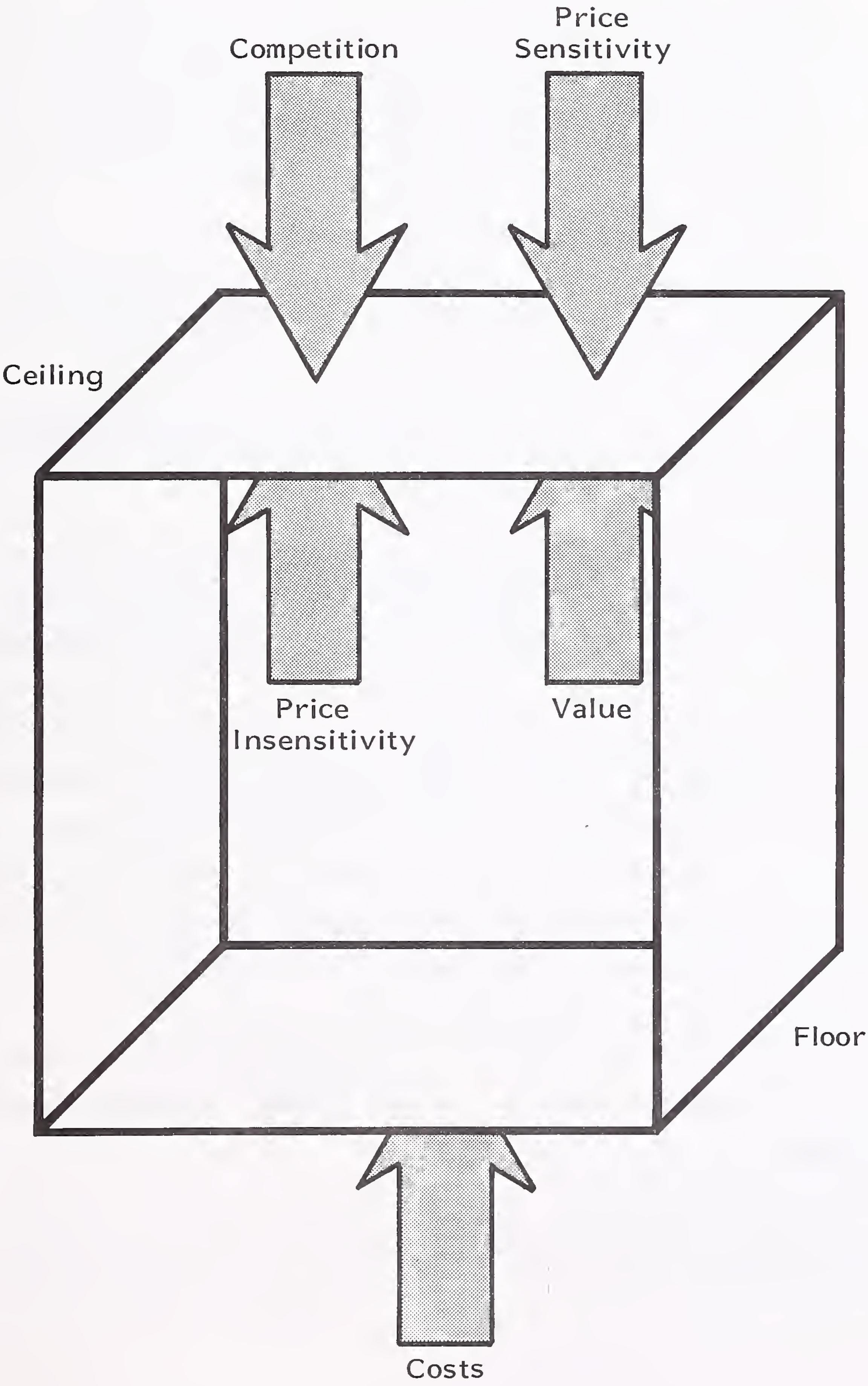
  - Improving performance/adapting to new operating environments.

  - Training and consulting.

- Vendors could make fairly precise projections on what customers would expect to receive in the last three categories. Software maintenance could then truly be sold.

EXHIBIT V-2

THE MOVABLE PRICING BOX



- A further step would be to unbundle each constituent (or group of constituents) and sell them separately.

  - This would be especially attractive for training and consulting. Currently, this is the largest demand area of the software maintenance organization. However, this demand is denigrated and termed "customer misuse". This is because it is usually "free" and, therefore, there are no rewards for supplying it.

  - To be economic, most training and consulting cannot be supplied live on a one-to-one basis. New approaches will be needed.

- Live, one-to-many seminars and presentations would be more economic, but unwieldy and still not cheap.

    - Videoconferencing would be more responsive, but it will be years before most customers will have the necessary facilities.

    - The best bet is video-based training (now offered by firms like ASI & Deltak). New developments in computer-controlled interactive video-based training will make these new training methods much more attractive and effective. They will be much more expensive to create, especially during the take-off phase in the technology.

- Training, probably in conjunction with an established training firm, would be a two- or three-stage process. Taking financial application systems as an example:

  - The first stage would review general financial principles and systems.

  - The second stage would focus on a particular industry and its special operational requirements.

  - The third stage would show how the package met these requirements, under differing circumstances, and how individual needs were met by particular features (and vice-versa).

- These interactive materials could also be used in a slightly modified form to supplement and, perhaps, supplant live hotline personnel.

C. HARNESSING TECHNOLOGY

- Software support and maintenance are second only to marketing for being labor-intensive. This labor intensiveness not only adds to costs but also threatens product quality; e.g.:

  - Relying on people to provide hotline information and training often prevents customer questions from being answered, either correctly or at all.

  - Identifying and fixing software problems besides being time consuming, is no guarantee that a new error will not be created. Software testing is at best only partially automated, and is all too often short circuited to save time and money.

- The next generation of interactive training devices should go a long way to upgrade customer training and problem resolution. A better, standard product would be supplied at what would ultimately be a cheaper price; initially, costs would be about the same.

- Maintaining the actual software will be a much more difficult area for applying technology.

  - There has been only limited progress to date in using technology effectively to develop new software (see INPUT's multiclient study, Improving the Productivity of Systems and Software Implementation). Most software development problems are in the unautomated areas of management, planning, and software design.

  - The productivity issues in software maintenance are at least an order of magnitude more difficult. They are not, however, unsolvable.

- For most vendors, the major productivity advance has been to use some type of on-line tool for coding (TSO, et al.). These speed up by perhaps 50% an activity that is no more than 10% of the new development process and considerably less of the maintenance cycle.

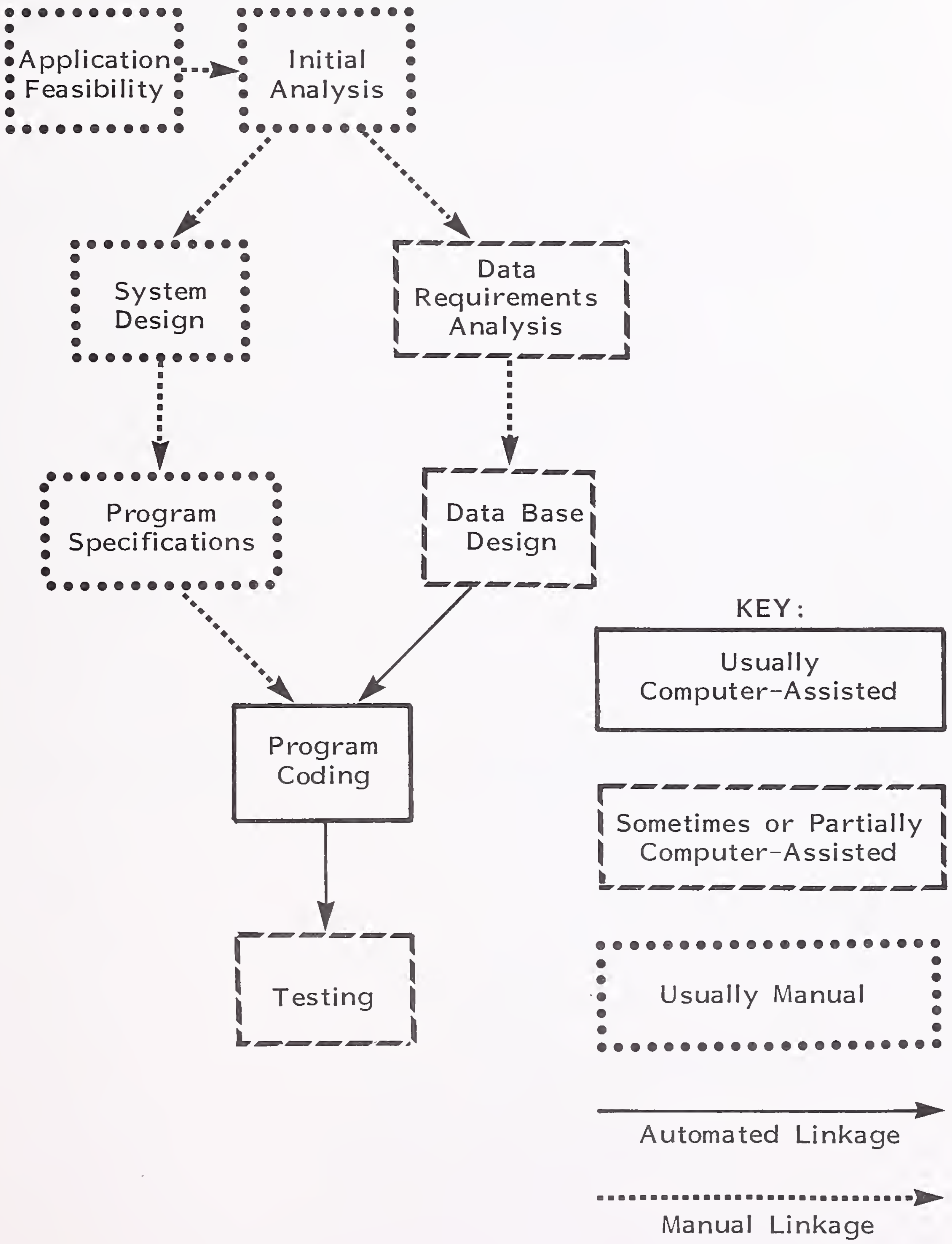
- Software maintainers would gain a powerful tool if they had an integrated development and support environment. At present there are few software tools that can usefully be applied to maintenance. Software maintainers are caught in a cleft stick:

  - No existing software development methodology provides an integrated software development and support environment, as shown in Exhibit V-3. Proposed approaches may rectify this, however (see, for example, INPUT's September 1982 Report, Business Graphics: Boon or Boondoggle?).

  - However, most of the benefits will accrue to the post-implementation phase (i.e., after software installation). This has greatly hindered acceptance of present tools. The weapons systems community has come closest to recognizing these problems, and the Department of Defense is sponsoring (and will soon require the use of) the ADA Soft-

EXHIBIT V-3

CONVENTIONAL SOFTWARE DEVELOPMENT PROCESS



ware Support Environment package of tools. It will be years, though, before this has an impact on the maintenance workload.

- Vendors should use such tools as are available now to the fullest extent possible. There will be a major long-term impact on the cost effectiveness of software maintenance. Exhibit V-4 lists selected current software productivity tools.

EXHIBIT V-4

SELECTED SOFTWARE PRODUCTIVITY TOOLS

- Constantine/Yourdon/Demarco Structured Design
- Jackson's Program Design Methodology
- Warnier-Orr's Structured Design
- Gane & Sarson's Structured Design
- Ross's SADT
- IBM's HIPO

APPENDIX A: QUESTIONNAIRE

1. Which of the following software product categories does your company offer?

PRODUCT CATEGORY	NUMBER OF MAJOR PRODUCTS	NUMBER OF INSTALLATIONS
Operating systems, operating system enhancements, utilities		
Computer operations support, performance measurement and improvement		
Data retrieval/reporting, DBMS		
Communications software		
Software development (compilers, optimizers, productivity tools)		
Applications software: Industry specific (list industries)		
Applications software: Cross industry		
Accounting		
Planning/forecasting		
Other (list)		

2. I would like to understand what your organization includes under the term "software maintenance." For each of the following functions, tell me whether each one is always, usually, sometimes, rarely or never considered to be part of software maintenance. (READ LIST AND FILL IN EACH LINE.)

2a. Software Maintenance Definitions

	PART OF MAINTENANCE*	EXAMPLE*
Fixes to programming errors		
Conversion to run on different hardware		
Conversion to run under different operating system		
Making existing features work better		
Extending existing features		
Adding new features		
Adding interfaces to other software products		
Training		
Consulting		
Other (describe)		

* WHERE "USUALLY, "SOMETIMES" OR "RARELY" ARE GIVEN AS ANSWERS, HAVE RESPONDENT GIVE AN EXAMPLE OF WHAT WOULD BE INCLUDED AND EXCLUDED FROM MAINTENANCE.

2b. Has your firm's definition of software maintenance changed in the past several years?

( ) YES ( ) NO

- If YES, please explain:

____

____

2c. Do you expect your definition to change in the next several years?

( ) YES ( ) NO

- If YES, please explain:

____

____

3a. What type of software problems or maintenance occur most frequently? Approximately what percentage is each and is the trend up or down?

PROBLEM	PERCENT	TREND

3b. Do you believe that software customers generally (not just your customers) are satisfied with the type of software support and maintenance they are receiving?

( ) YES ( ) NO

- Why?

  ____

  ____

4a. I would like to get a better understanding on how your organization performs software maintenance. For each of the functions below could you tell me the name of the organization unit responsible and the number of full time and part time employees assigned? (If you know the full-time equivalents (FTE), please tell me that instead.)

FUNCTION	ORGANIZATION UNIT NAME	NUMBER OF PEOPLE		
		FULL TIME	PART TIME	FTE
Determining the cause of errors				
Fixing errors				
Enhancement analysis and design				
Coding and installing enhancements				
Revision of documentation				
Telephone contact with customers on software maintenance				
Personal contact with customers on software maintenance				
Other (describe)				

4b. What problems, if any, has your organization had with your internal organizational arrangements for software maintenance?

____

____

____

____

4c. What changes, if any, are planned in organizational responsibilities or reporting relationships?

____

____

____

____

4d. To what extent does the software development organization become involved in software maintenance?

____

____

- How satisfactory is this?

  ____

  ____

- What changes, if any, are planned in their involvement?

  ____

  ____

5a. How do you decide the amount of resources (dollar, manpower, etc.) that will be devoted to software maintenance?

____

____

____

5b. Is there a formal budget for software maintenance?

( ) YES ( ) NO

- If YES, how effective has this been in managing the software maintenance function?

  ____

  ____

- How easily can the budget be modified in the course of the year?

  ____

  ____

- Is software maintenance a profit and loss center?

  ( ) YES ( ) NO

  - If YES, what are your most profitable and least profitable elements?

    Most: ____

    ____

    Least: ____

    ____

5c. Are resources shifted back and forth between software development and maintenance?

( ) YES ( ) NO

o If YES, describe the process?

____

____

____

____

5d. Do you plan to change the level of resources devoted to software maintenance?

( ) YES ( ) NO

o If YES, by what percent? ________%

o Why?

____

____

o How will this affect service levels?

____

____

5e. What aids do you now use for software maintenance?

____

____

- What has been their impact (please quantify, if possible)?

  ____

  ____

- What other aids do you expect to use in the future?

  ____

  ____

6. How important is each of the following methods to your firm for identifying software problems and potential enhancements? How important do you think each will be in the future? Why? (5 = High Importance, 1 = Low Importance)

METHOD	IMPORTANCE		REASON
	NOW	FUTURE	
Salesforce Feedback			
Support Staff Feedback			
User Groups			
Surveys			
Hot Lines			
Field Visits By Software Staff			
Analysis of Problem Reports			
Other (Describe)			

7. I would like to understand the process used to get software revisions to customers.

7a. How often do you release revisions to software products?

____

____

To what extent does this differ by product? (Describe)

____

____

7b. What percent of the time do the customers apply the software revisions themselves? ________%

IF LESS THAN 100%: Who applies the revisions in other cases and why?

____

____

7c. How frequently do customers have problems that can't be handled by the usual software revision cycle?

____

____

o What causes this?

____

____

____

- o How are these cases handled?

____

____

____

- o Are these exception cases increasing or decreasing? Why?

____

____

____

8. To what extent do you use telecommunications to either identify software problems ("remote diagnostics"), solve them or automatically download software revisions? How much do you expect this to increase in the future? Why?

TELECOMMUNICATIONS USAGE	AMOUNT USED*		REASON
	NOW	FUTURE	
Identify software problems			
Solve software problems			
Download software revisions			

* ALWAYS, USUALLY, SOMETIMES, RARELY, NEVER; OR PERCENT.

9. How price conscious is the typical customer of software maintenance? (5 = High Price Consciousness, 1 = Low Price Consciousness) ________

- o How do you know?

____

____

- o Are customers becoming more or less price conscious?

____

____

10a. What is the general process for determining the pricing of software maintenance?

____

____

____

____

10b. How important is each of the following for your firm in determining software maintenance pricing? (5 = High Importance, 1 = Low Importance)
(READ LIST AND ASK REASON)

CRITERIA	RATING	REASON
Profitability on resources used		
Percent of current software price		
Value to customer		
Competitors' pricing		
Other (describe)		

11. What percent of your firm's overall software revenues now come from software maintenance?

__________%

o What do you expect this percent to be in five years?

__________%

- Why? ____

____

- IF UNWILLING TO ANSWER ABOVE: What do you expect the average annual growth rate of your firm's software revenues to be over the next five years? __________%

Why? ____

____

12. Do you feel that software in general has become more reliable in the last five years?

( ) YES ( ) NO

o Why?

____

____

____

o What has been your firm's experience?

____

____

____

o What kind of measurements do you perform to verify reliability?

____

____

____

13. Do you feel that software <u>in general</u> has become more maintainable in the last five years?

( ) YES ( ) NO

- o Why?

  ____

  ____

  ____

- o What has been your firm's experience?

  ____

  ____

  ____

- o What kind of measurements do you perform to verify maintainability?

  ____

  ____

  ____

14. What percent of your software maintenance revenues would you say are derived with little or no active selling on the part of your firm? ________%

o Is this percent increasing or decreasing? ____________

- Why? ____

____

o Where you do have to actively sell software maintenance, what type of techniques do you use?

____

____

- Which organizational unit(s) does the actual selling? ____________________

____

o How much more potential do you feel there is for marketing software maintenance as a product?

____

____

15. What do you feel are the most important issues currently concerning software maintenance? Why?

____

____

____

____

____

____

16. What changes do you think will have occurred in software maintenance in, say, five years?

____

____

____

____

____

____

17. Which of your software support services do you find that your customers are most receptive to? Why?

____

____

____

____

18. Could you send me:

- o Description of your major software products?

  ( ) YES ( ) NO

- o Software product and maintenance price lists and sample contracts?

  ( ) YES ( ) NO

APPENDIX B: VENDORS INTERVIEWED

APPENDIX B: VENDORS INTERVIEWED

- Ten hardware companies, with total revenue ranging from $55 million to $4 billion, were interviewed.

- Twenty-seven hardware companies, with total revenue ranging from $4 million to $125 million were also interviewed.

- Multiple interviews were conducted with some companies to obtain information. The typical respondent had the title of Director or Vice President.

APPENDIX C: THIRD-PARTY SOFTWARE MAINTENANCE

APPENDIX C: THIRD-PARTY SOFTWARE MAINTENANCE

- Building up a business in third-party software maintenance would be much more difficult even than third-party hardware maintenance, which has market entry problems of its own; i.e.:

    - Finding skilled technicians.

    - Covering key geographic areas.

    - Keeping up with product changes.

    - Overcoming customer resistance to leaving the original vendor.

- In many cases, of course, hardware vendors will willingly devolve part of their maintenance business for certain product lines and/or geographic areas.

    - This often makes sense since there are relatively few economies of scale in traditional on-site repair.

    - Vendors can even face diseconomies of scale without a critical mass of business.

- Software maintenance is quite different in several key respects:

    - There are few on-site calls.

- Fixes to known errors generally need be made only once.

- A deep understanding of a software package is usually required, an understanding not easily gained by outsiders.

- Much maintenance, actual and potential, is actually enhancement. The maintenance and development functions cannot be completely separated.

- Customer resistance to third-party software maintenance would be even greater than for third-party hardware maintenance.

- For these reasons, third-party software maintenance is generally not technically feasible and will in any case be uneconomic for buyer and seller.

APPENDIX D: THE SOFTWARE MAINTENANCE ENVIRONMENT

APPENDIX D: THE SOFTWARE MAINTENANCE ENVIRONMENT

A. GENERAL SOFTWARE ISSUES

- It is important to understand how software maintenance relates to software in general.

- The software development process consists of the following steps.

  - Analysis.

  - Design.

  - Coding.

  - Testing.

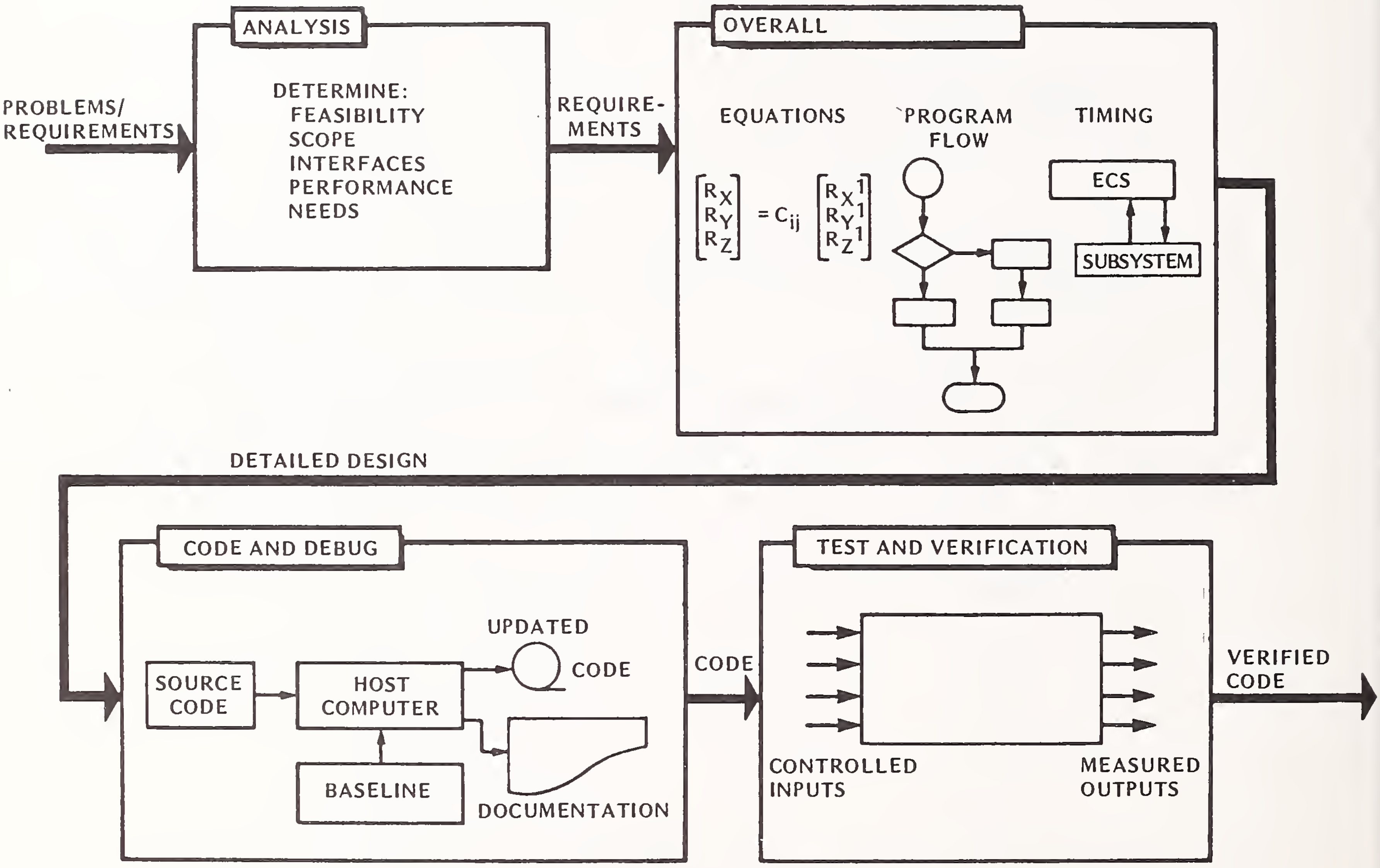
- This is shown in more detail in Exhibit D-1.

  - The conceptual content is identical whether new development, enhancements, or error corrections are involved.

  - This general process is equally applicable to vendors and IS departments.

EXHIBIT D-1

THE SOFTWARE PROCESS



- From a life cycle planning and cost standpoint, there are two phases:

    - Development (containing 30% to 40% of costs).

    - Maintenance (including error correction and some enhancements, and containing up to 60% to 70% of costs).

- These percents are based on reasonably reliable data. Unfortunately, much software cost data is very soft. Historic data, nonstandardized and not consistently gathered, often lacks linkages between project event reporting and cost reporting. (This is discussed in more detail in the next section of this chapter.)

    - Cost estimating tools are in their infancy.

    - Recent studies for the Air Force have shown that cost estimates for the average software projects undertaken for them were 60% of the actual figure.

        - The variances were nonlinear; i.e., correction factors would not help.

        - Different software environments and applications make the same model behave differently, sometimes erratically.

    - These kinds of problems are not because of a lack of estimating models. Some of the most discussed models are shown in Exhibit D-2.

        - Most models now focus on the early (development) part of the life cycle.

EXHIBIT D-2

SELECTED SOFTWARE ESTIMATING SOURCES

- PRICE Software Lifecycle Model (RCA PRICE Systems)
- Quantitative Software Management, Inc. (L. Putnam)
- Department of Defense Micro Estimating Procedure
- SOFCOST, Grumman Aerospace
- Doty Associates
- Tecolote Research, Inc.
- Boeing Computer Services Model
- TRW Model (Wolverton)
- Aerospace Corporation Model
- USAF, Electronic Systems Command

        - Maintenance resource estimating is subject to enormous variation.

- There are various tradeoffs between developing new software and enhancing existing software. One view is presented in Exhibit D-3.

    - These curves should only be taken as suggestive, given the lack of reliable software costing data.

    - It is obviously in the interests of those who would maintain software (and make a business of doing so) to devise methodologies to keep the maintenance cost curve from rising.

B. VENDOR-SPECIFIC ISSUES

- Vendors have additional software issues to be concerned with:

    - Software products as part of an overall marketing strategy.

    - Whether to bring out a new feature as an enhancement to an existing product or as a new product.

C. RELATION OF HARDWARE AND SOFTWARE

- There has been a trend for software costs to increase relative to hardware costs. Exhibit D-4 shows the general relationship over time. Different installations can be at different points on the curve at any time, depending on their needs, applications, and sources of software.

EXHIBIT D-3

SOFTWARE DEVELOPMENT - MAINTENANCE TRADEOFFS

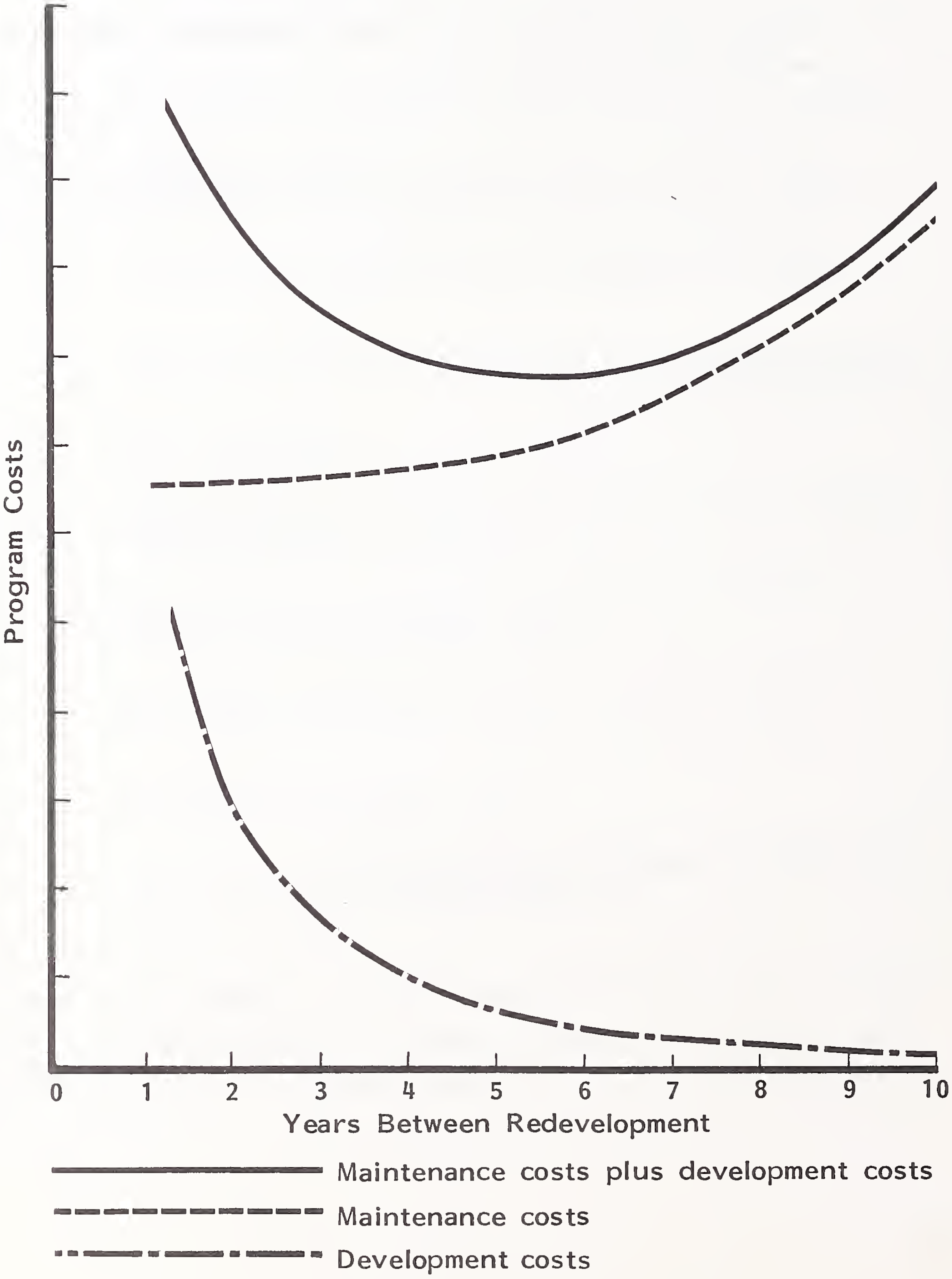
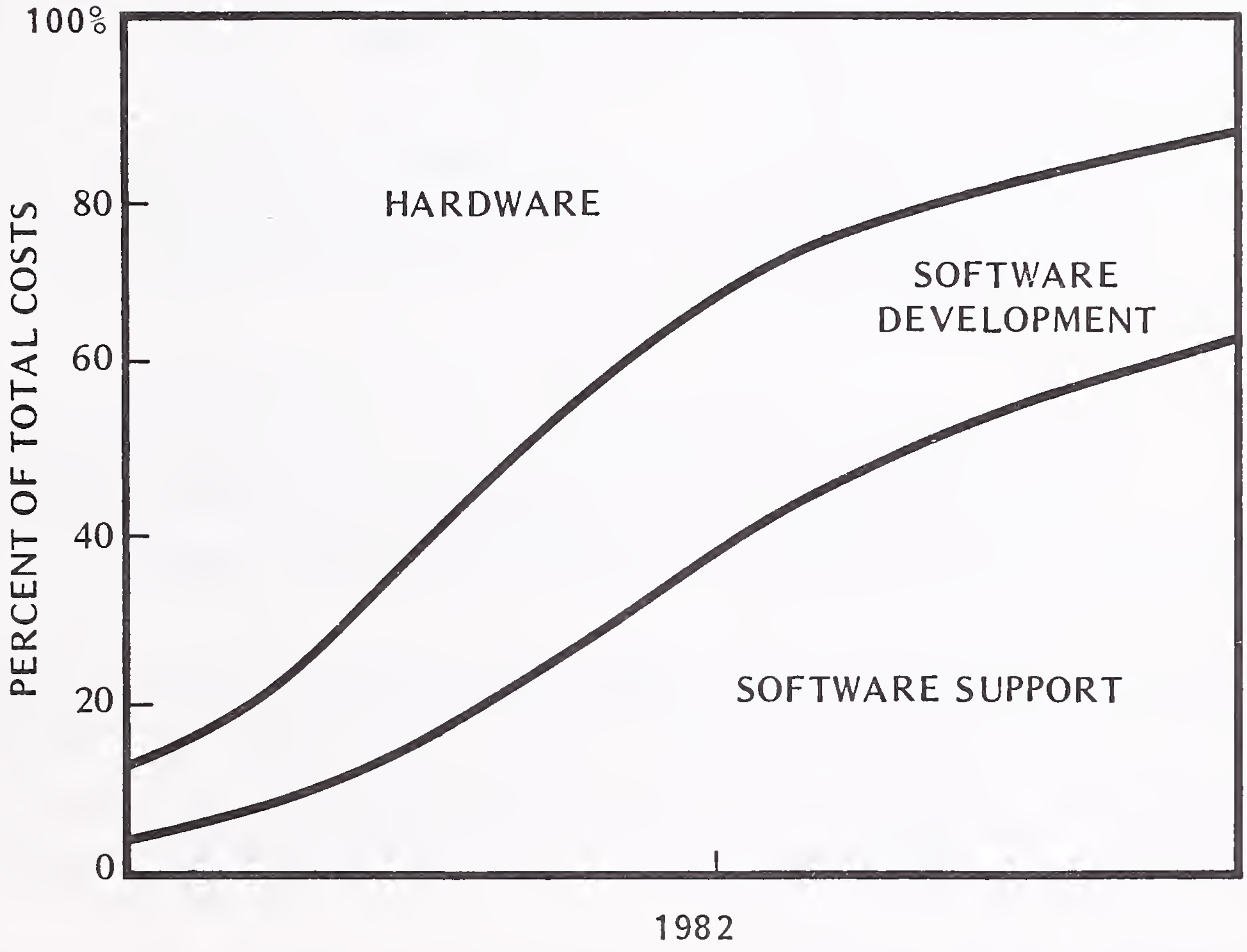


EXHIBIT D-4

GENERAL HARDWARE/SOFTWARE COST TRENDS



(Adapted from Barry Boehm)

- Hardware costs often fall in absolute terms, and almost always in relation to processing accomplished.

- There have been very few analogous advances in software productivity.

● It is useful to contrast the hardware and software life cycles as is done in Exhibit D-5. Many parts of the two processes are similar.

  - As a result, those with a hardware background sometimes underestimate the magnitude and importance of the real differences between the hardware and software cycles.

    . In software, the design/development phase is critically important; there is nothing analogous to the manufacturing phase.

    . The maintenance phase is much more important in software, not only because more initial bugs may appear, but also because software is far more enhanceable than hardware.

    . Software may become nonfunctional, but it doesn't wear out. In principle, software is eternal; in practice, it is still usually so machine- and application-dependent that it pays to discard rather than rework it. (This may change in the future.)

● Hardware design and manufacturing is logical and scientific. Software design and building is still much more of an art than a science. Scientific design principles are in their infancy. Software construction is a handicraft. the increasingly used term, software engineer, is a misnomer.

  - These observations apply to almost all new software development.

  - Software maintenance is far, far behind new development.

EXHIBIT D-5

HARDWARE/SOFTWARE LIFE CYCLE COMPARISONS

	HARDWARE	SOFTWARE
Development	Determine User Requirements	Determine User Requirements
	Develop Product Concept (Functional Design)	Develop Product Concept (Functional Design)
	Specify Component Design (Detailed Design)	Specify Component Design (Detailed Design)
	Build and Test Prototype	Implement and Test Programs*
	Develop Manufacturing Techniques	--
Manufacturing/ Installation	Manufacture Product	Copy Programs*
	Make Product Available to and Train User	Make Program Available to and Train User
Maintenance/ Improvement	Maintenance (Correct Component Failures)	Maintenance (Correct Implementation and Design Errors)*
	Recall Product to Correct Design Flaws	Maintenance (Provide Enhanced Capabilities: Adapted to Changed User Environment)
	Enhance Product	
Phase-Out	Unit is Unusable and Unrepairable (Replace with New Unit)	Software May Migrate to New Hardware Generation*
	Product is Obsolete	Product is Obsolete

* MARKS IMPORTANT DIFFERENCES

D. THE SOFTWARE MAINTENANCE ENVIRONMENT

- The general software maintenance process is similar in principle to the software development process in terms of the sequence of the tasks and the general skills required.

- However, there are significant differences between software development and maintenance. There is a common misconception that software maintenance demands fewer skills than does software development.

- This is not true; if anything, software maintenance may be more demanding in general than development; e.g.:

  - Support is often necessary for multiple versions of software (e.g., where packages have been semi-customized).

  - Unforeseen requirements must be coped with.

  - There are often severe capacity and performance constraints.

  - The most important software maintenance will be carried out in a crisis environment.

- While the requirements are often more demanding than in the development area, the resources to carry them out are often inferior.

  - Software maintainers must work in an obsolescent environment.

  - Documentation is often poor, sometimes nonexistent.

- Documentation is often poor, sometimes nonexistent.

- Productivity tools are even less advanced than in the development area.

- It is not surprising that maintenance can be very expensive. According to one study, while the cost per line of new code averaged $75, maintenance could cost as much as $4,000 per line of code.

  - This is because the later in the life cycle that a program is changed, the more expensive the change becomes, as shown in Exhibit D-6.

  - Not only do previous steps have to be retraced, but sometimes the design of the system itself will no longer be appropriate for the newly required function.

E. KNOWLEDGE REQUIRED FOR SOFTWARE MAINTENANCE

- Three different levels of knowledge are required to perform satisfactory software maintenance:

  - Technical knowledge.

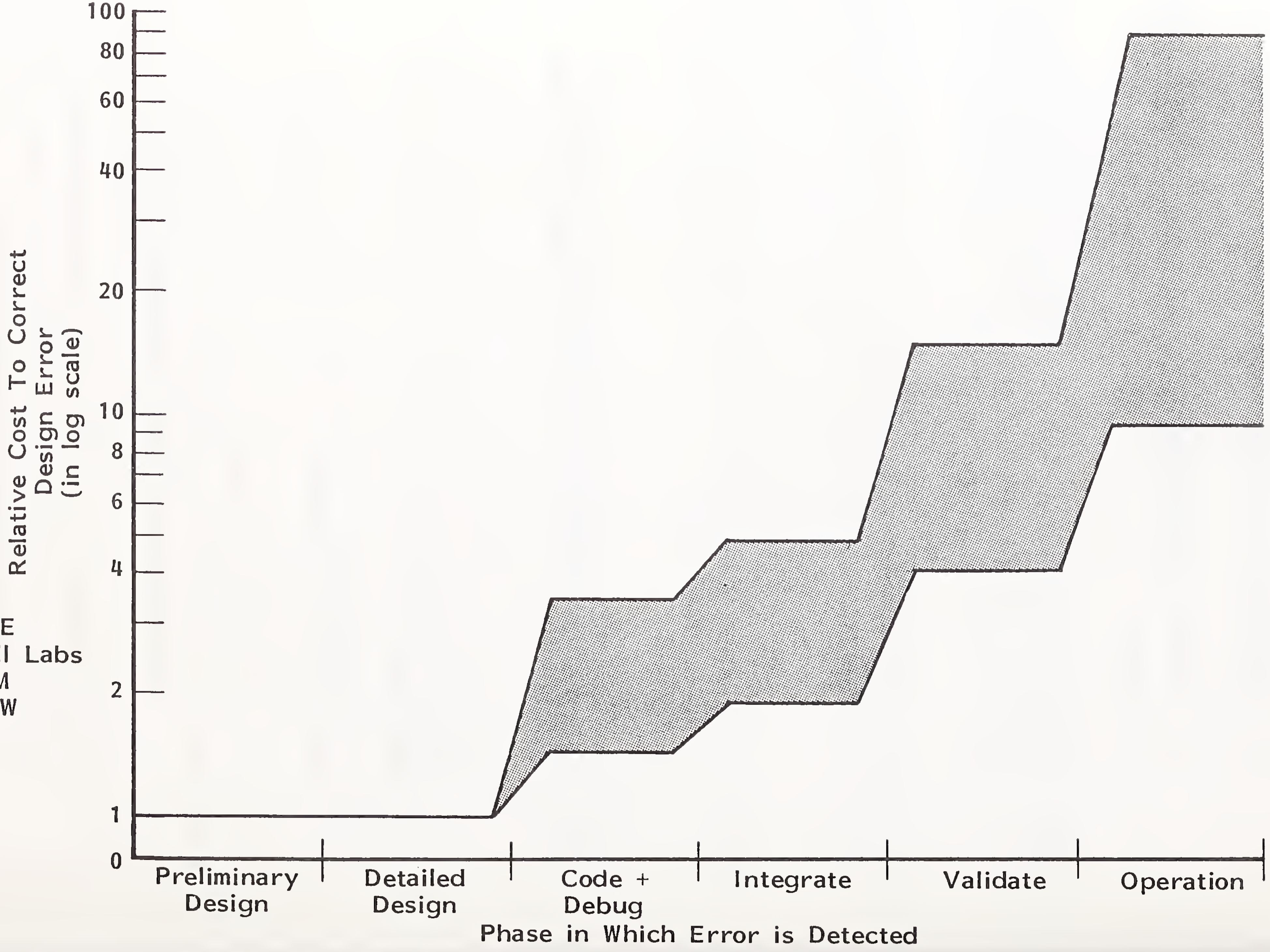
  - Functional knowledge.

  - Application knowledge (for application software).

- Currently, there is considerable confusion among these three levels. There is a tendency to focus on technical knowledge and issues, in large part because of the obvious, critical problems to be resolved.

EXHIBIT D-6

SOFTWARE MODIFICATION COSTS



Sources: GTE
Bell Labs
IBM
TRW

- However, there is also an element of comfort involved in dealing with familiar technical issues.

- Exhibit D-7 shows what is, in INPUT's view, the proper relationship among the three levels of knowledge.

- Technical knowledge focuses on software methodology; i.e.:

  - Software system design/modification.

  - Software module design/modification.

  - Coding and languages.

  - Software problem identification/resolution.

    - Pre-operation = debugging, testing, and verification.

    - Post-operation = maintenance.

  - This knowledge resides in computer programmers and technical assistants. The skill levels required may range from low to high.

- Functional knowledge is nonapplication related, but includes both software and, increasingly, hardware issues.

  - Examples include:

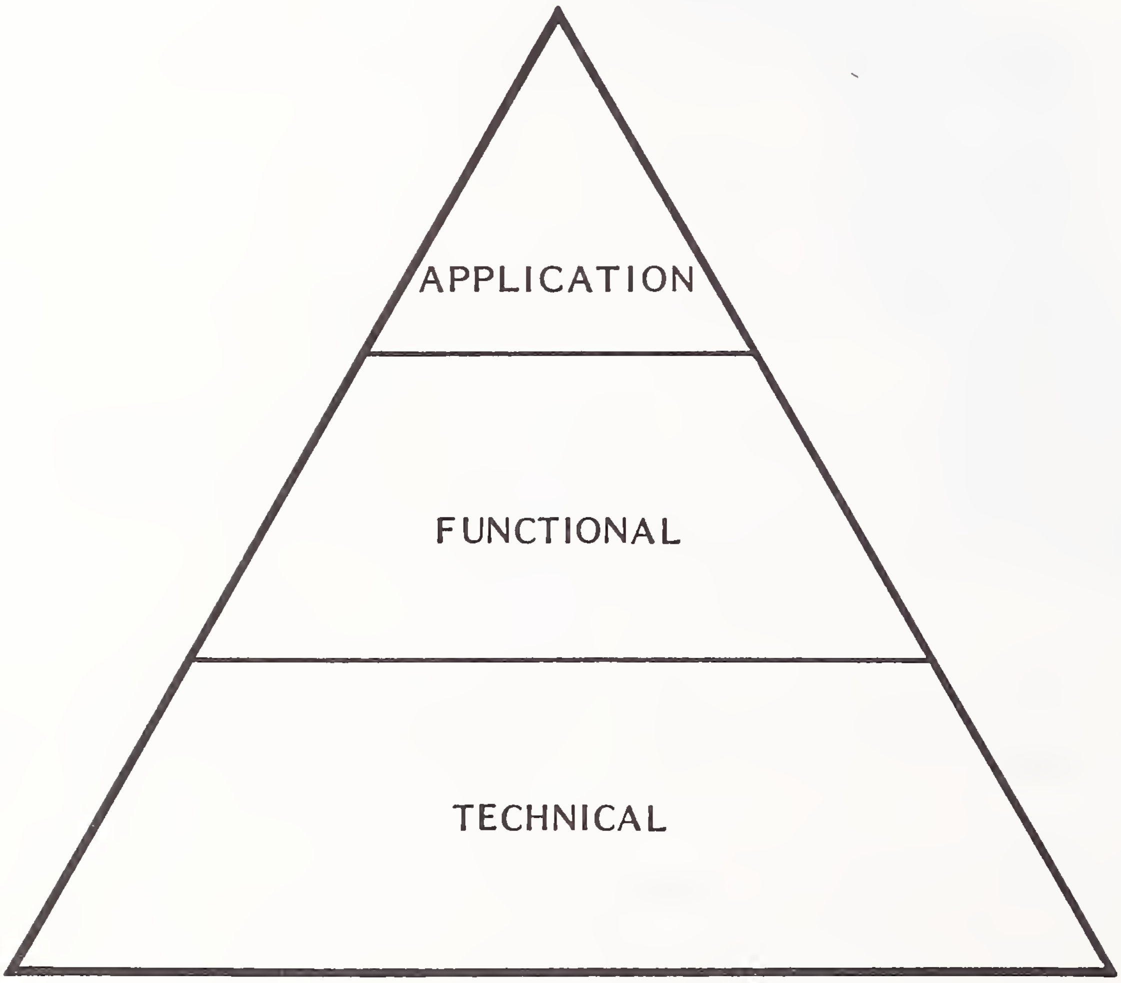
    - Data base management.

    - Networks.

    - Distributed data processing.

EXHIBIT D-7

KNOWLEDGE LEVELS



        - Security.

    - This knowledge resides in computer scientists. Required skills range from medium to very high.

- Application knowledge builds on technical and functional knowledge. It is critical to understand the ultimate purpose of an applications package to use the package effectively.

    - Examples include:

        - Banking systems.

        - Insurance systems.

        - Manufacturing systems.

        - Financial systems.

    - This knowledge resides in applications specialists who should, however, have enough technical knowledge to communicate and understand the technical specialists.

F. CURRENT SOFTWARE MAINTENANCE PROBLEMS

- The division of responsibility is now often unclear among the field, marketing, development, and maintenance organizations for identifying and making changes to software.

- The process is very labor-intensive.

    - Methodological approaches are uncertain.

- Scarce and expensive skills are often diverted to coding and clerical functions.

- Relatively few tools exist. What does exist is of variable quality/utility. In any case, tools are not standardized.

- The result is a high-cost, sometimes unsatisfactory product.

G. THE SOFTWARE MAINTENANCE IMAGE

- It is almost impossible to overstate the horrible image attached to software maintenance.

  - "Garbage picker" might be appropriate.

  - There is a strong view, held by both developers and maintainers, that no advancement is possible, and that only inferior people get involved with maintenance. A software maintainer does simple-minded work and does not improve professionally.

- The reality usually is bad.

  - Maintenance personnel often work with obsolescent software.

  - The reward for success is being stuck in a particular application indefinitely.

  - There is no obvious career path (except out of maintenance).

- There is no body of knowledge or tools to make the work easier or more professionally fulfilling.

- While the objective requirements of software maintenance are at least as demanding as in development, this is not always perceived by those assigning or performing the work. The result is unsatisfactory performance in terms of one or more of the following:

  - Quality of deliverables.

  - Cost.

  - Time required.

- Some organizations are more successful than others in improving both the image and product of software maintenance (the two are closely related):

  - Some will call it something else or avoid calling it anything ("X" System Group, rather than "X" System Maintenance Group). This can be surprisingly effective, but is often difficult to do in large organizations which prefer clarity to obscurity in titles.

  - Others purposely mix development and maintenance within the same group. This works even where maintenance is a very high proportion of work done.

  - The most interesting case is the systems programming staff in customer organizations, which in reality performs almost pure maintenance. Yet this function is very high status, for a mixture of reasons:

- Systems programmers deal with the heart of the computing system. The mystique rubs off.

- In the past, the job was, in fact, much more creative than it now is. It used to be common to make operating system changes for a particular installation.

- Systems programmers are viewed as a repository of technical knowledge (and often are). This confers status.

- The case of the systems programmers is not directly applicable to other software maintenance. However, there are lessons that should be applied.

- In general, though, the exceptions merely prove the rule that software maintenance is looked down on and is often ineffective.

  - There is high turnover.

  - Much of the work is costly and ineffective.

  - There is little sense of direction or view of software maintenance as an organized discipline.

APPENDIX E: HYPOTHETICAL SOFTWARE PRODUCT: TEN-YEAR HISTORY

APPENDIX E: HYPOTHETICAL SOFTWARE PRODUCT: TEN-YEAR HISTORY

ASSUMPTIONS

- Peak sales in year six.

- Introductory price intentionally low; raised to market price in year three; 10% annual increase until year nine.

- No charge for maintenance first year after sale; thereafter, assumed that all customers are under maintenance.

- Annual maintenance cost is 12% of the sales price in the same years.

EXHIBIT E-1

TEN-YEAR SOFTWARE PACKAGE AND MAINTENANCE REVENUE FOR A HYPOTHETICAL PRODUCT

YEAR	UNITS SOLD	PRICE/ UNIT ($ thousands)	PACKAGE REVENUE ($ thousands)	UNITS UNDER MAINTENANCE	MAINTENANCE PER UNIT ($ thousands)	MAINTENANCE REVENUE ($ thousands)	TOTAL REVENUE ($ thousands)	MAINTENANCE PERCENT OF TOTAL
1	20	$10	$ 200	0	0	0	$ 200	0
2	60	15	900	20	$1.8	$ 36	936	4%
3	100	18	1,800	80	2.2	176	1,976	9
4	150	20	3,000	180	2.4	432	3,432	13
5	200	22	4,400	330	2.6	858	5,258	16
6	250	24	6,000	530	2.9	1,537	7,537	20
7	250	26	6,500	780	3.1	2,418	8,918	27
8	200	28	5,600	930	3.4	3,162	8,762	36
9	200	30	6,000	1,130	3.6	4,068	10,068	40
10	200	30	6,000	1,330	3.6	4,788	10,788	44

INDEX

INDEX