NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

MICROCOMPUTER MICROECONOMICS

by

Mark Alan Doernhoefer

December 1982

Thesis Advisor: Dan C. Boger

Approved for public release; distribution unlimited

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
Microcomputer Microeconomics

5. TYPE OF REPORT & PERIOD COVERED
Master's Thesis
December 1982

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Mark Alan Doernhoefer

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Naval Postgraduate School
Monterey, California 93940

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Naval Postgraduate School
Monterey, California 93940

12. REPORT DATE
December 1982

13. NUMBER OF PAGES
64

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release, distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Microcomputer economics, economic evaluation, microcomputers, microeconomics, econometrics

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This paper outlines the procedures for an economic analysis of a microcomputer system designed for personal use. It provides a methodology for application identification and quantification of the benefits derived from the applications. Empirical decision rules are suggested for the key decisions of quantity and mix of software, start-up and cessation timing, and resource allocation. These rules are based upon analysis of marginal opportunities gained and f regone over the lifecycle of the system. Particular emphasis is

placed upon the role of software in the economics of the system. Economies of scale and sensitivity analysis are also discussed. The thesis serves as a structured beginning for further research into microcomputer system modelling.

Approved for public release; distribution unlimited.

Microcomputer Microeconomics

by

Mark A. Doernhoefer
Lieutenant, United States Navy
B.A., University of Missouri, 1976

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
December 1982

ABSTRACT

This paper outlines the procedures for an economic analysis of a microcomputer system designed for personal use. It provides a methodology for application identification and quantification of the benefits derived from the applications. Empirical decision rules are suggested for the key decisions of quantity and mix of software, start-up and cessation timing, and resource allocation. These rules are based upon analysis of marginal opportunities gained and foregone over the lifecycle of the system. Particular emphasis is placed upon the role of software in the economics of the system. Economies of scale and sensitivity analysis are also discussed. The thesis serves as a structured beginning for further research into microcomputer system modelling.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

A. BACKGROUND

In 1971 Intel Corporation constructed a general purpose microprocessor chip in response to a request from Datapoint for a front end processor for their terminals. Due to the long lead time involved with the Datapoint application and with Datapoint's permission, Intel began selling the chips on the open market. Two versions were created, the 4004, a basic microprocessor, and the 8008, an upgraded model. Market response was sluggish and Intel began to lose interest in the whole project. However, in 1975 Altair packaged Intel's 8008 in kit form with the S-100 bus and created the first true microcomputer, the Altair 8800. The following year Apple Computers (which currently holds 20% of the microcomputer market) was born in a garage in Cupertino, California. Three other companies also introduced their flagship microcomputer models that year. For various reasons the latter three units were not supported by consumers and are no longer in production. Commodore, Heath, and Radio Shack all introduced their inaugural microcomputers in 1976.

The industry took off during 1978-1979. Dozens of firms began marketing their own systems. Other companies, large and small, introduced their own peripheral equipment compatible with the more popular systems. Software vendors offering both operating systems and applications programs virtually eliminated the need for micro users to learn programming. Currently *Byte* magazine [Ref. 1: p. 446] reports that IBM expects to sell one million Personal Computers by the end of 1984 and Apple estimates there are

400,000 Apple II models in use. Apple's sales have leveled off at approximately 20,000 systems per month. The new sixteen-bit microcomputers will place the power of a minicomputer in a desktop-sized package.

Awed by the technology and surprised at the low cost, users have crowded the computer stores in a rush to computerize their businesses. Neophyte computer owners discover that they have failed to adequately analyse their situation before investing. Many are forced to use systems that are saturated from the start; others do not have enough applications to fill the top-of-the-line systems they have purchased; and still others spend excessive amounts of time deciphering masses of data generated by poor software. These people are learning the lessons which mainframe owners learned over the past three decades. They have succumbed to the diseconomies of microcomputers.

B. THESIS OBJECTIVES

1. Decision Rule Development

The purpose of this paper is to provide a set of economic decision rules for the evaluation of a microcomputer system for a single user. Actual decisions are generally made within the context of a limited and highly approximate abstraction of the actual situation. The decision rules contained herein provide structure to these abstractions. The use of the rules enable comparison between competing computer systems. The same conditions and assumptions are extended uniformly across all systems which permit value measurement and allow the potential user to rank the systems on an interval scale. Naturally, microcomputer selection cannot be made solely on an economic basis, but an analysis using these techniques can provide input to the overall decision process.

2. Use Analysis

This paper will attempt to provide a systematic method for the identification of microcomputer applications for a single user in a small business environment. It will discuss how to extend the computing resource throughout a large organization; in other words, how to determine which levels and individuals of an organization should be allocated personal computers. More importantly, it will attempt to identify and quantify the benefits to be derived from the implementation of selected applications. Not surprisingly, micros are being marketed as just another piece of office equipment. However, unlike most office equipment, microcomputers have a set of applications from which the user can select specific uses depending upon the software accompanying the unit. The first step in system planning is identification and evaluation of prospective applications. The application identification method that will be presented is designed to be flexible and expandable so as to be of use to all persons seeking cost-effective uses of personal computers.

3. Effectiveness

Another goal of this thesis is the analysis of effective microcomputer use. A system can be perfectly efficient in that it processes information at the lowest possible cost but it may not be very effective. A principle element in system effectiveness is the quality of information it produces. The characteristic of quality as it applies to information is a composite of its content, age, accuracy, and importance [Ref. 2: p. 16]. The simplest way to improve information quality is to "clean up" input or output format. For example, the yellow pages of the telephone book contains higher quality information than the

white pages if the user is seeking a list of all computer dealers in the geographic area. The information is more efficiently formatted with respect to the desired task. Producing effective information on output is largely a function of the software employed. The tradeoffs in software cost and information quality will be considered later in the software costing discussion.

C. ASSUMPTIONS

This economic analysis is based upon several assumptions which form a framework for the decision rule. Microcomputer systems will be the only computers under consideration. For discussion purposes a microcomputer is defined as an eight or sixteen-bit machine with no more than one megabyte of main memory that costs less than $10,000 including peripherals. At this writing there are approximately one dozen sixteen-bit micros in production. Although manufacturers have not finalized their price structure for the new machines, the methodology will be useful as soon as prices are set. The use of the Intel 8088 and 8086 microprocessors with their one megabyte memory address space in these second generation micros surpasses the main memory capacity of many of the more expensive minicomputers. Therefore, an arbitrary limit of $10,000 will be established to help distingush between micros and minis. This is necessary as the minicomputer software market is radically different from the micro software market.

1. Organizational Assumptions

For purposes of this evaluation it will be assumed that sufficient funds are available in the organization to purchase the system. In addition, only hardware purchase will be considered. There are a myriad of lease and

lease-to-own plans available, far too many to consider here. Furthermore, these plans are highly sensitive (much more so than price) to dealer oversupply, new product introduction, and interest rate fluctuations. Computing services provided by an external vendor will not be considered either. Haftka and von Mayrhauser [Ref. 3: p. 7] point out that the buy or contract decision is highly dependent upon the charging algorithm used by the service bureau. Their survey included five service bureaus and found a wide variance in computing costs for the execution of a benchmark program.

2. User Assumptions

Finally it will be assumed that the system must be bundled as a turnkey operation for a naive user. Unlike the large computer operation, the small systems user does not have the time to learn computer operation or the funds to support a software or operations staff. The user, with the help of off-the-shelf software, software and system documentation, and dealer support, should be able to treat the computer as a black box that accomplishes the desired application. Programming, programming languages, operating systems, and networking protocols are all beyond the scope of this user.

D. METHODOLOGY

While this effort is not meant to be a treatise on microcomputer procurement, it is important for the reader to understand the overall method of computer purchase. This is necessary in order to see how the techniques discussed in later chapters fit in to the purchase plan. There are many different procurement strategies. Gupton [Ref. 4: p. 202], Barden [Ref. 5: p. 87], and Lu [Ref. 6: p. 36] all recommend various plans for effective small system selection. Each of

these methods have strong and weak points. Gupton's method, for example, contains a very thorough discussion of application analysis but does not say much about costing. Whatever methodology under consideration, operation of this decision aid requires three additional steps; application identification, cost analysis, and modeling.

During the application identification phase a search for areas of use which would benefit the user is conducted. This paper will suggest a structured procedure for benefit analysis. This step will yield a set of benefits which will form the basis for the rest of the evaluation. The second phase, cost analysis, examines the hardware-software costs incurred to accomplish the desired applications. This is not merely an exercise in shopping, but a detailed review of the incremental costs involved in implementation. Modelling, the final step, is the comparison of benefits identified in step one with the costs of the system discovered in step two. Use of the models will also elucidate start and stop times, resource allocation, and tradeoffs.

These steps are designed to be repeated with different cost and benefit elements to produce the optimum system. The model acts as an impartial measurement tool which helps cope with the complexity of the decision. The results of each iteration will call attention to areas where additional savings may be realized. As discussed above, this methodology is not meant to replace standard decision making techniques, but rather it is designed to augment good business practice. These calculations do not consider intra-organizational environmental factors and, thus, cannot evaluate system feasibility. For example, cash flow analysis of the optimum system can help the decision maker determine whether the purchase will fit in with his long range cash management plan.

II. BENEFIT ANALYSIS

A. FRAMEWORK FOR BENEFIT DETERMINATION

Due to the low cost of microcomputer systems and the retail store approach to marketing micros, hardware salesmen often make grandiose promises regarding system performance. Unsophisticated buyers are led to believe that a microcomputer will solve all their business problems. This passage from Nahil [Ref. 7: p. 7] is illustrative of some of the claims made to entice prospective customers.

> Small business computers can help you:
>
> 1. Cut costs.
> 2. Increase productivity.
> 3. Improve efficiency.
> 4. Make sounder decisions.
> 5. Help business grow.

This cost of generalization of benefits to be achieved from microcomputer use has no meaning to the economist. In order to demonstrate actual value of a system, benefits must be quantified with respect to each individual application. Since the micros are designed to be used by one person, the personal utility of the benefits must be taken into account.

1. Market for Benefits within the Organization

Within each organization there exist areas from which automation will provide economic payoffs. Each area must be identified prior to the system design to enable the designer to focus his efforts on optimizing the performance of that functional area. Problem definition is of utmost

import as illustrated in this passage from the Infotech Report on Computing Economics [Ref. 8: p. 87].

> When a manager tackles any problem, there is an inherent assumption that the problem being tackled is relevant to the objectives of the organization concerned. This is an obvious point but one that is frequently missed. No amount of management skill applied subsequently will help if the initial choice of project is wrong or of marginal significance.

The effort to computerize must be concentrated on the specific applications that will produce the greatest return. This is particularly true when dealing with mini and micro systems where the computing resource may be limited. Inherent in this concept is the realization that a tradeoff point may be reached where it is beneficial to discard the idea of a minicomputer and consider a mainframe.

There are many ways to select applications. Needs statements, requirements definitions, and other methods of application identification do not consider the costs involved. A manager may require the system to perform an uneconomical application and then wonder why his computer system productivity is so poor. Sharpe [Ref. 9: p. 9] states:

> Cost/effectiveness analysis is very much at variance with another approach to decision making that can best be termed the 'requirements' approach. The latter recommends that the decision-maker (1) determine his requirements and then (2) find the cheapest way to satisfy them. Such a procedure, if followed literally, can lead to optimal decisions only by chance. Indeed the concept of a requirement or need is completely foreign to an economist. Firms 'need' the biggest and best computer available. Researchers 'require' an almost unlimited amount of computer time with the very highest priority. Central processors 'need' a large number of peripheral devices to ensure that they will be used to capacity. In short, needs are either unlimited or so large they can hardly ever be met in practice.

This trap is easily avoided when working with microcomputers. Since each microcomputer is selected for one

individual's personal use, the applications can be tailored to his specific needs. The key to successful economic analysis is comparison of the cost benefit relationship incurred with each sucessive application.

2. Cost Element Approach

The first step in designing an effective microcomputer system is the identification of specific applications to be considered. Application descriptions must be highly detailed in order to select the appropriate hardware-software combination. Wickham [Ref. 10: p. 51] states:

> With these simple , low cost microcomputer systems, the business manager can minimize risks and costs by deploying these small increments of computer power against specific, well defined problems in his business. Using the computer for a single application such as job costing, payroll, or inventory provides quick solutions to the real problem areas without creating new major problems.

By focussing the computing power on the precise application, the decision maker increases the probability of system success. It also provides him with some insight as to the operation of the systems within his organization.

The military has developed a set of cost elements for determining costs of procurement projects. All of the possible elements comprising system lifecycle costs are grouped in a logical order to provide a tool in overall cost analysis [Ref. 11]. It is apparent that this method could also be used in identifying potential benefit areas in an existing organization. To perform the analysis, the decision maker lists all cost centers within his organization. Table I is an example of a cost element listing. It is critical that the elements be listed in as small an area as possible so that proper software can be obtained. Software packages contain many functions. A package called "The

TABLE I

Partial Cost Element Structure

- 1. Administration
  - 1.1. Personnel
    - 1.1.1. Payroll
      - 1.1.1.1. Payroll tax preparation
    - 1.1.2. Employee records
  - 1.2. Finance
    - 1.2.1. Cash management
    - 1.2.2. Invoicing
    - 1.2.3. Accounting
      - 1.2.3.1. Tax accounting
      - 1.2.3.2. Financial accounting
      - 1.2.3.3. Bookkeeping
  - 1.3. Word Processing
    - 1.3.1. Text editing
    - 1.3.2. Electronic mail
    - 1.3.3. Electronic filing
- 2. Logistics
  - 2.1. Inventory
    - 2.1.1. Stock rotation
  - 2.2. Equipment
    - 2.2.1. Maintenance
    - 2.2.2. Resource management
    - 2.2.3. Procurement planning
  - 2.3. Transportation
- 3. Operations
  - 3.1. Sales and Marketing
    - 3.1.1. Economic forecasting
    - 3.1.2. Account development
  - 3.2. Production
    - 3.2.1. Cost control
    - 3.2.2. Job costing
    - 3.2.3. Research and development
    - 3.2.4. Production tracking
  - 3.3. Training

Accountant", for instance, may contain tax and bookkeeping elements all of which may not be needed.

B. QUANTIFICATION OF BENEFITS

It is very difficult to place a dollar value on the benefits generated from microcomputer use. Automation yields savings in time, improved output accuracy, and the ability to use sophisticated analysis techniques which were

not feasable prior to the purchase of the computer. These benefits can be illustrated on a small scale by considering the use of a pocket calculator to balance a checkbook. The time savings and accuracy improvement in this case are obvious but now the superior computational ability of the calculator makes it easy for the user to forecast his future balance and budget accordingly. What was it worth for that person to be able to plan his budget? Chances are a dozen people would give a dozen different answers. These are some of the problems that must be considered when attempting to quantify benefits.

1. Revenue Generation

Time savings can be expressed as a dollar value by estimating the amount of revenue that can be generated over the amount of time saved. In addition, some actual costs such as paper and record storage costs can be reduced. These savings have a multiplier effect if the savings are reinvested in revenue producing projects. It is important to note that the computer system itself does not raise revenue or profits except in the case of a computer service bureau. One example of this method of benefit quantification is the case of a travel agent. If the agent can reduce the service time of each client by using a microcomputer to automatically print airline tickets the agent can service more customers. The increased revenue resulting from the time saved by not typing tickets by hand is the benefit resulting from this application.

2. Cost Savings

Benefits can also be quantified by calculating the labor and material costs saved. This is particularly useful if actual outlay for part-time or non-salaried workers is involved. Reduction of bookkeeper hours by automated

bookkeeping is an excellent example of this method of benefit evaluation. Although this procedure is much simpler than the revenue generation scheme, the benefits may not be directly translatable into return on invested capital. Ross, writing in the Infotech Report on Computing Economics [Ref. 8: p. 100] relates the case of the BOAC. The airline showed it cost twice as much to make a plane reservation after computerization. However, due to automation they were able to fly more people on each flight. The net result was a 22 percent annual return on a $150 million investment. This account points out the need to consider both methods and judge all downstream effects of automation.

C. PROBLEMS IN BENEFIT QUANTIFICATION

Problems generated in benefit quantification fall into two major groups, benefits which cannot be measured and benefits which may or may not accrue. Items in the former category should be listed for subjective consideration by the decision maker. If he feels these benefits are worth the cost, they may be assigned an arbitrary dollar value and factored into the decision rule. Benefits which are in doubt can be assigned a probability of occurance and multiplied by the amount which could be anticipated to yield an expected value. Here again the decision maker must exercise caution as the accuracy of these calculations is apt to be very poor.

1. Benefits Which Cannot be Measured

One of the primary unmeasureable benefits is the improvement of information accuracy as a result of automation. This element is very dependent upon the quality of input data; the familiar "garbage in-garbage out" principle. This complicates the decision maker's subjective evaluation

of information accuracy. In service-oriented businesses, microcomputers can help businessmen improve customer relations by enabling the employees to deal with customers quickly and efficiently. By keeping the customer accounts on line, employees can talk intelligently with customers instead of having to look it up and return the call. Errors in accounts can be corrected on the spot. Other benefits which cannot be measured are those affecting productivity, efficiency, and business growth cited earlier by Nahil.

2. Uncertain Benefits

Forecasting and prediction may or may not produce benefits, however the payoffs from advance information gained by forecasting can be excellent. For example, if a businessman can correctly predict a market trend and stock accordingly, he can reap huge profits. Muller [Ref. 12: p. 12], in his evaluation of small business micro applications mentions the many sophisticated numerical analysis techniques that micros are making available to small business. He also comments on the spreadsheet type of program which enables the novice to answer many of his "what if" questions or, in other words, perform sensitivity analyses. Many of the benefits derived from these applications can improve the decision-making power of the user. All of these uncertain benefits involve situational or external factors which complicate evaluation.

D. UTILITY

The value of the personal computer is highly dependent upon the values of the person using it. All of the quantification schemes discussed above must be tempered by the decision maker's utility function. Figure 2.1 shows the indifference curves formed by plotting one benefit versus

another. Each of these curves represent lines of equal utility, or, "iso-utility curves". The expansion path that the individual will follow as utility increases travels in the direction of the arrow. It represents the decision maker's willingness to exchange one benefit, or application, for another as the overall system utility increases. It is important to note that this path may not necessarily be a straight line. Each individual may have different tradeoffs at different levels of utility.

Utility must also be considered when dealing with costs and benefits. Jones [Ref. 13: p. 9] defines costs as "disutility producing objects" and has plotted costs and benefits on indifference curves. The slope of these indifference curves is negative and measures the individual's tradeoff between costs and benefits. Jones calls this slope the "rate of psychological cost benefit substitution". It measures the individual's willingness to attempt to gain additional benefits at extra cost. Both these utility considerations translate into a set of ratios or weightings that must be applied to the different benefit levels to reflect the personal choices of the user.

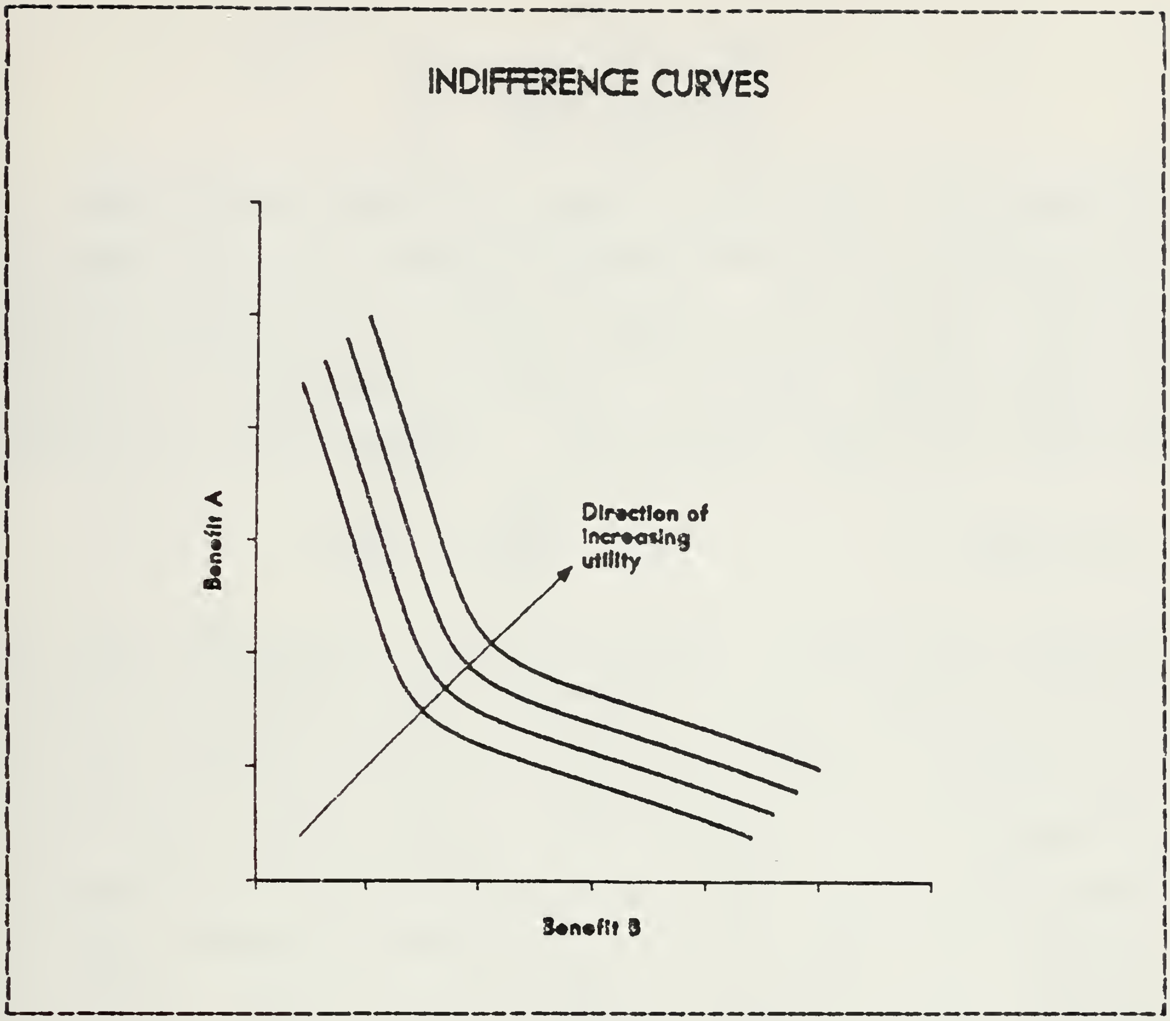


Figure 2.1 Utility and Indifference Curves.

III. COST ANALYSIS

The costing disussions which follow will be concerned not only with the direct costs but also the incremental or marginal costs of the various items. The three relationships of cost-value, marginal cost-marginal value, and net value are illustrated in Figure 3.1 from Sharpe [Ref. 9], where q* is the optimum. The marginal costs are required

$$MC = \frac{dC}{dq} \qquad \text{(eqn 3.1)}$$

for the calculation of optimum levels of inputs and outputs. Since the marginal cost is defined as the change in total cost (C) brought about by a one unit change in output (q), the derivative equation for marginal cost is given by equation 3.1, where output is defined as the amount of computation. The term value will refer to the dollar amount of the benefits qualified in the previous chapter. Therefore, the total value (TV) line in the upper graph in Figure 3.1 tracks the increased value of the benefits created by microcomputer use. So, as total value increases, the marginal value (MV) is the change in total value for a

$$MV = \frac{dV}{dq} \qquad \text{(eqn 3.2)}$$

one unit change in computation (output). This is illustrated in equation 3.2 . Obviously, the user wishes to gain the maximum value at the minimum cost. This is equivalent to maximizing the net value, total value minus total cost. Since the goal is to maximize net value (NV = V - C), the optimum level occurs when the marginal net value is zero. Consider equation 3.3, which shows that the optimum can also

$$\frac{dNV}{dq} = \frac{dV}{dq} - \frac{dC}{dq} = 0 \qquad \text{(eqn 3.3)}$$

be found by setting marginal costs equal to marginal value. Figure 3.1 provides a graphic illustration of these relationships. In the upper graph, the point of greatest separation between total cost and total value corresponds to the point in the center graph where the marginal costs and marginal values are equal. The lower graph shows that at this point, net value is maximized.

These relationships are important in that they form the logical basis for determining the decision rules discussed below. Each element of the microcomputer system will be discussed separately in this chapter. In determining the optimum levels in this chapter, all of the costs and values associated with the various elements will be analyzed with respect to the marginal cost-marginal value relationship. The individual element optimal level will occur at the point where the marginal costs attributed to that element equal the marginal values derived from the element. In Chapter IV, all the elements required by the system will be combined to form the formal system decision rule.

A. SOFTWARE COSTING

The goal of software costing is to identify each small area of benefit and tie the cost of software to these applications. Wickham [Ref. 10] recommends deploying computers like chess pieces:

> With these simple, low cost microcomputer systems, the business manager can minimize risks and costs by deploying these small increments of computing power against specific, well defined problems in his business. Using the computer for a single application such as job costing, payroll, or inventory provides quick solutions to the real problem areas without creating new major problems. ... Hardware prices are no longer considered

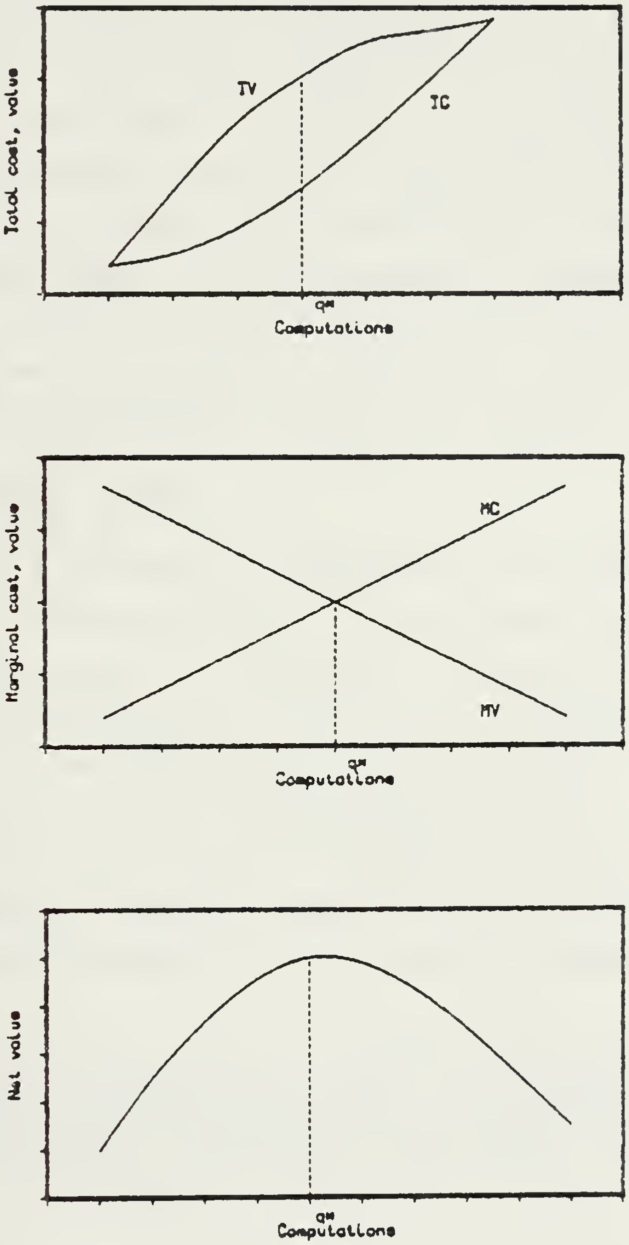


Figure 3.1 Total, Marginal, and Net Relationships at Optimum.

> to be a limitation of market growth. The major limitations at the present time on the expansion of the market for low cost business systems are marketing, user risk, and software costs.

The cost of software is variable in that the decision-maker can control the cost by controlling the number of applications. This variable nature of software costs is particularly true when the cost of hardware is viewed as a fixed cost distributed among all applications. This outlook is useful as it enables the separation of hardware and software costs and, thus, allows separate consideration of both elements. In fact, in the discussion of hardware cost, the case will be made to consider software as the only effective input to the microcomputer decision problem.

1. Software Market

Custom software produced by in-house programmers will not be considered here. It is simply not feasible for an individual using a personal computer to require the support of programmers and analysts. Instead the discussion will focus on the purchase of off-the-shelf, or standard, software. As with any standardized object, standard software may require the user to modify his practices to use the various programs. Wickham [Ref. 10: p. 51] states that the user finds modification of his business objectionable:

> The experience to small business suppliers indicates that standard software is usually not acceptable to the user without some modifications. The users tend to want to make the computer conform to their business procedures and methods. This naturally adds to the cost and risk of the new system.

Of course, the user will wish to tailor the microcomputer system to his specific operations. Unfortunately, the cost of software modification is quite high. The user must fight the urge to add additional cost to the software without

gaining major benefits. Indeed, the cost of a small modification to a computer program often makes the entire application economically infeasible. However, the standard software market is growing rapidly and there are many standard packages to select from. Careful software selection, aided by the application elements identified above, can enable the user to find the "best fit" of software to applications without major modifications to his routine.

At this point, a brief discussion of the standard software market is indicated. This discussion of the marketplace and the pricing problems faced by software houses is needed to help the user gain some insight to the environment surrounding software selection. The process of creating standard software is highly labor intensive. This creates an unusual market demand curve as illustrated in Figure 3.2 which assumes the software house is attempting to assign a price for a software package based on anticipated sales (demand). If a software developer estimates a small demand for the product, he will raise the price to attempt to recover his costs. If the demand for the product is widespread, he can lower the price to distribute development costs over the market. In addition to development costs, standard software houses also pay for the maintenance of their software. Updates are provided free or for a small additional charge to registered owners of their programs. Software prices have remained relatively constant over time unlike the huge price decreases in hardware. This, once again, is due to the labor intensive nature in software design and the lack of breakthroughs in increasing programmer productivity [Ref. 14: p. 47].

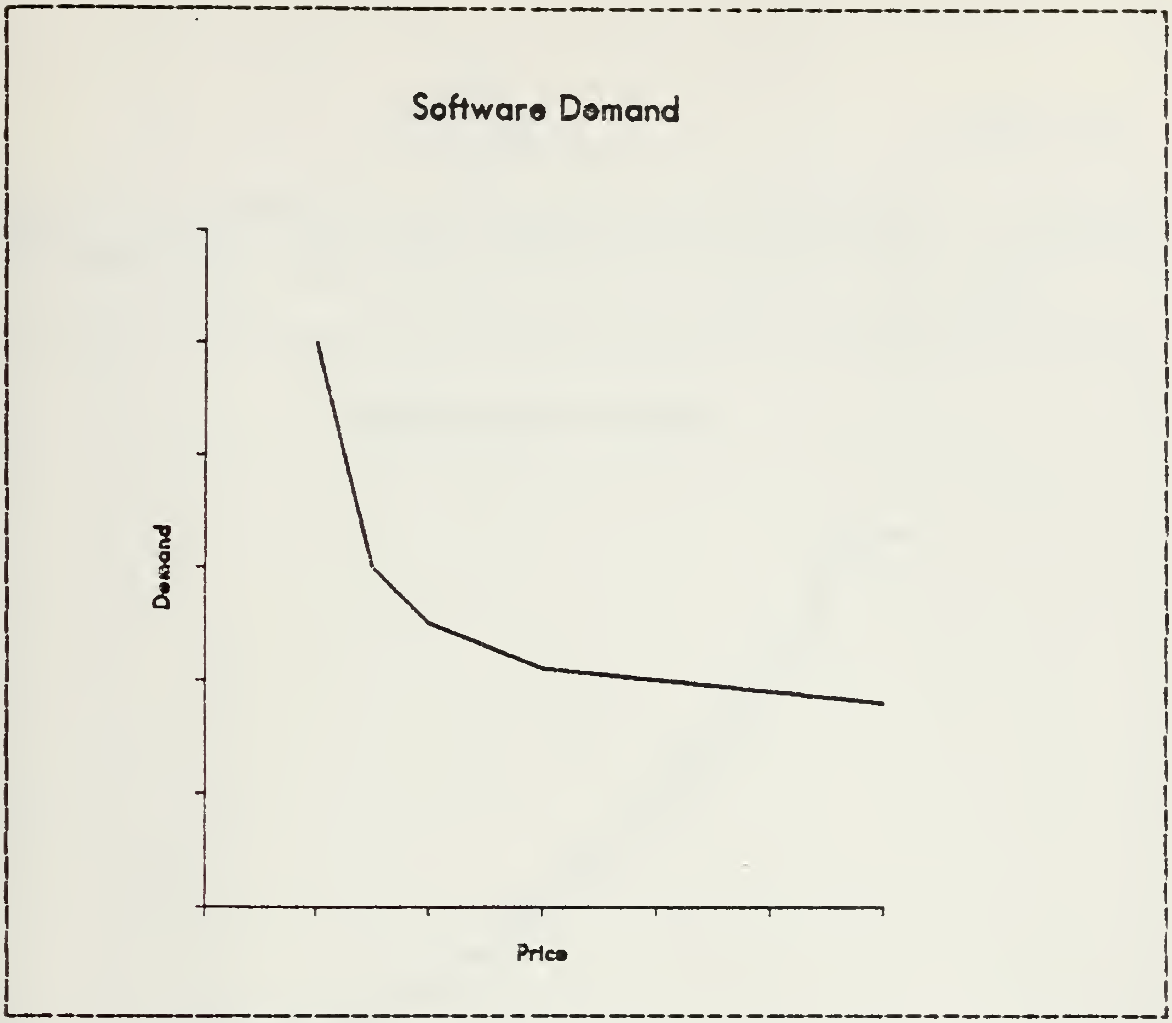


Figure 3.2 Standard Software Market Demand Curve.

2. Costing Method

The costs of software form a step function with respect to the number of applications involved. If c_i represents the cost of a piece of software for application i, the total software cost is represented by equation 3.4,

$$C_s = \sum_{i=1}^{N} c_i \qquad \text{(eqn 3.4)}$$

$$V=\sum_{i=1}^{N} v_i \qquad \text{(eqn 3.5)}$$

where N represents all applications considered. Likewise the values can be calculated as in equation 3.5, where v_i is

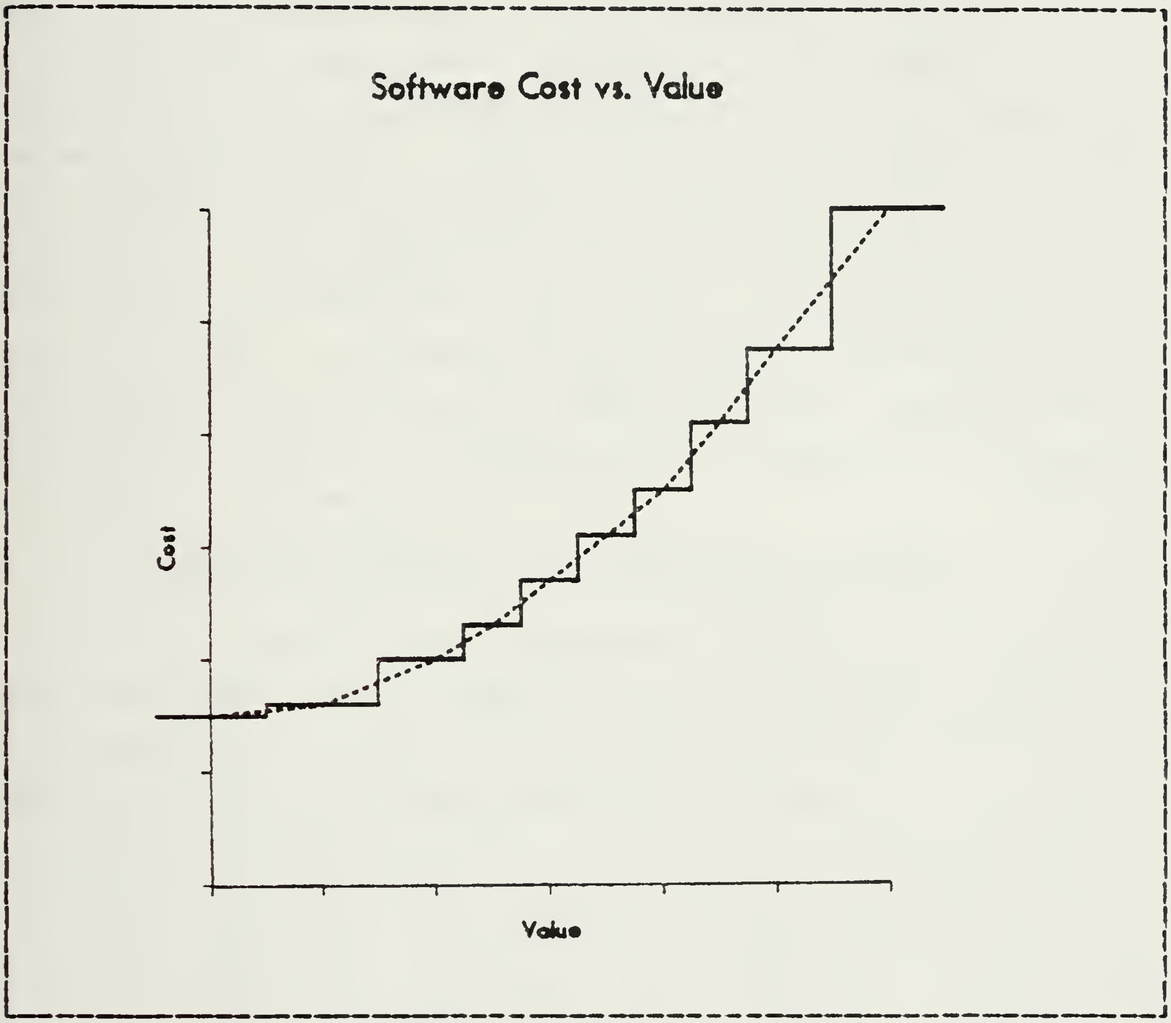


Figure 3.3 Software Cost Value Relationship.

the value of using one particular piece of software. This leads to the relationship of costs to values as depicted in Figure 3.3 . Note that no value is gained until the costs

of the first piece of software has been realized. The dashed curve drawn through the step function represents the progression of the cost-benefit relationship.

The next set of graphs, contained in Figure 3.4, illustrate the optimum value of software. The x axis, applications, is the number of applications or quantity (q_S) of software to be procured. This means that q* and i*, the optimum number of applications, within the step function, are the same. The upper graph shows the total cost of software (C_S) and the total value of software (V_S) versus applications. From the optimality equations above, the optimum amount of software is then demonstrated by equation 3.6 . In words, equation 3.5 says that the optimum quantity of software is that which sets the marginal net value to zero. The marginal net value, by definition, is the marginal value less the marginal cost, which indicates that these latter two values must be equal at the optimum quantity of software.

3. Software Selection and Information Quality

To round out the discussion on software costing, a few words must be said with regard to selection and information quality. As mentioned above, information quality is a measure of system effectiveness and, as such, is not considered in the decision rules. The software is responsible for handling data within the program and the effective display of the processed information. Without software that protects information quality, the value of the processed data is likely to decline. Naturally, this attribute of the software affects software price. Figure 3.5 from Emery [Ref. 2: p. 396] shows software cost as a function of information quality. It is a clear example of the law of diminishing marginal returns. It also underscores the need for the prospective software buyer to carefully select those

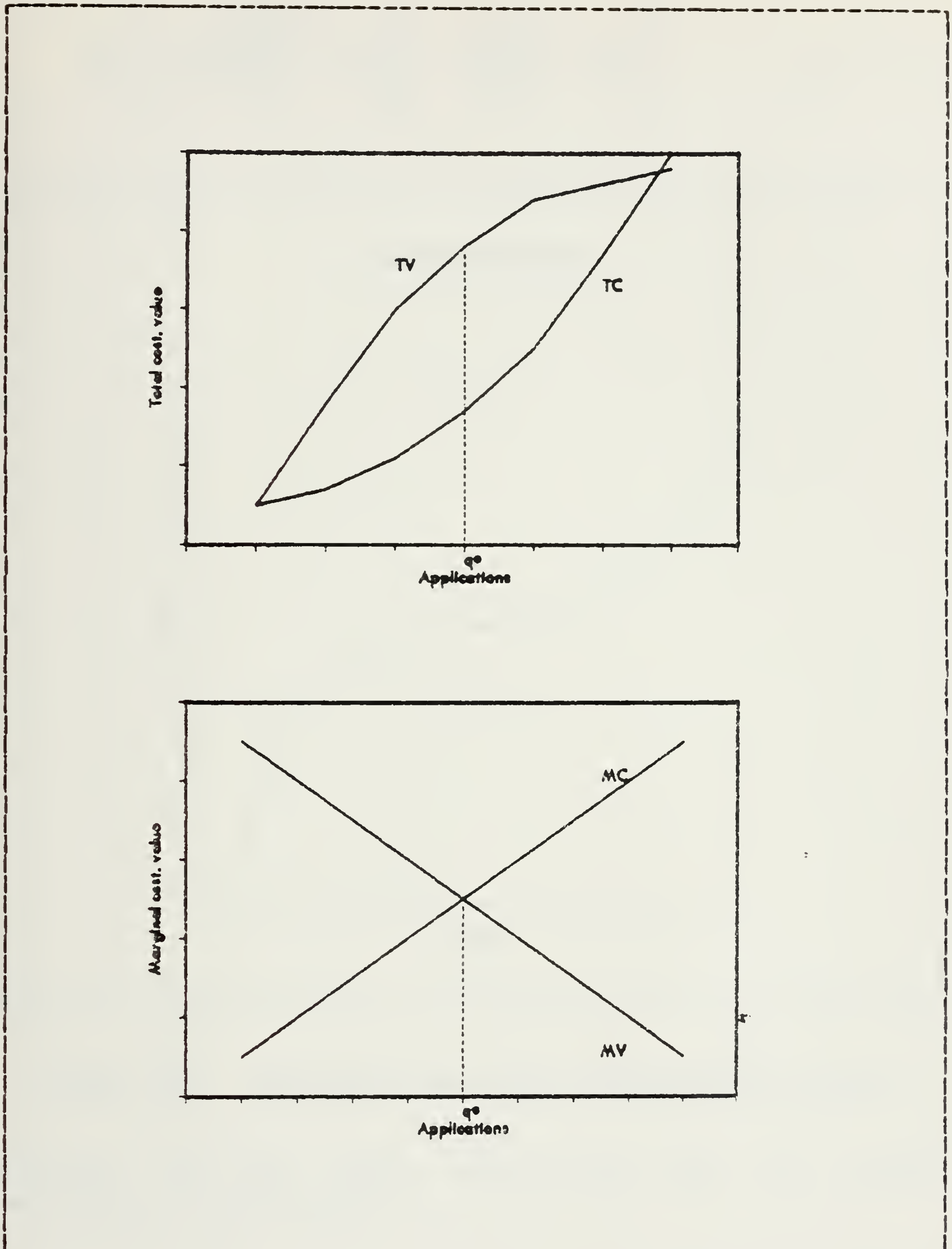


Figure 3.4 Optimum Software Relationships.

$$q_s^* = \left\{ q_s \left| \frac{dNV_s}{dq_s} = \frac{dV_s}{dq_s} - \frac{dC_s}{dq_s} = 0 \right. \right\} \qquad \text{(eqn 3.6)}$$

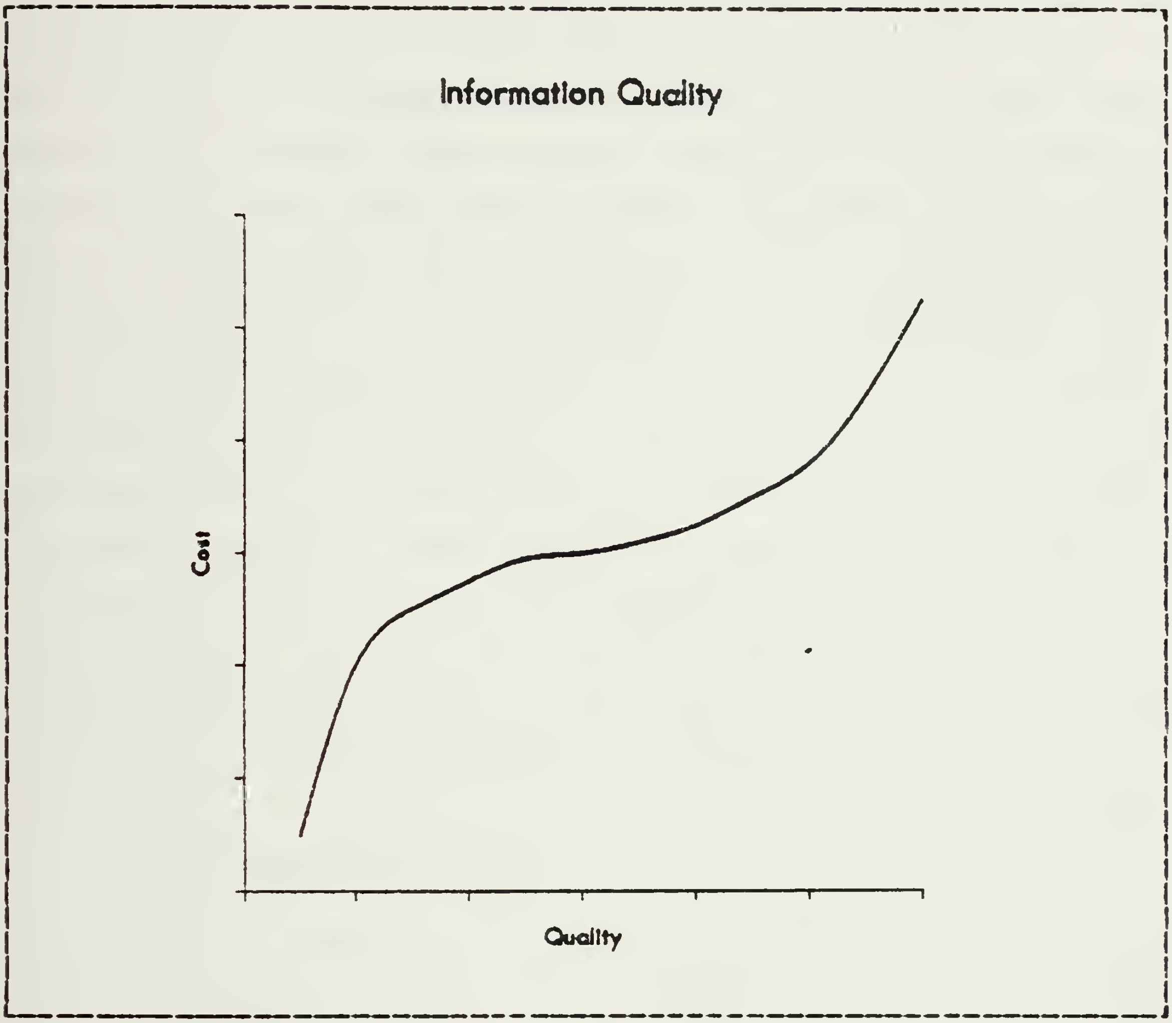


Figure 3.5 Software Cost versus Information Quality.

programs that will handle and present data in a manner befitting the application.

B. HARDWARE COSTING

Microcomputer costing differs from mainframe costing by more than just the orders of magnitude involved. Most micros are designed as a single circuit board ("motherboard") with expansion slots for additional features contained on cards. This structure enables the user to specify the particular features required and, more importantly, to exclude features not needed for his application. Large computers carry many standard features, some of which are not needed but cannot be eliminated for a reduced system price. This means that microcomputers can be adapted to a specific function more readily than mainframes which tend to retain their general purpose characteristics. It must be pointed out that the trend in large systems today is toward specialization and away from the general purpose mainframe. Database machines, large newspaper-type word processors, and industrial robots are a few examples of specialized mainframes. In addition, micros are limited in the amount of expansion possible while a mainframe usually has more slack before the saturation point is reached. This makes the mainframe more forgiving of design mistakes than the micro.

1. The Hardware Market

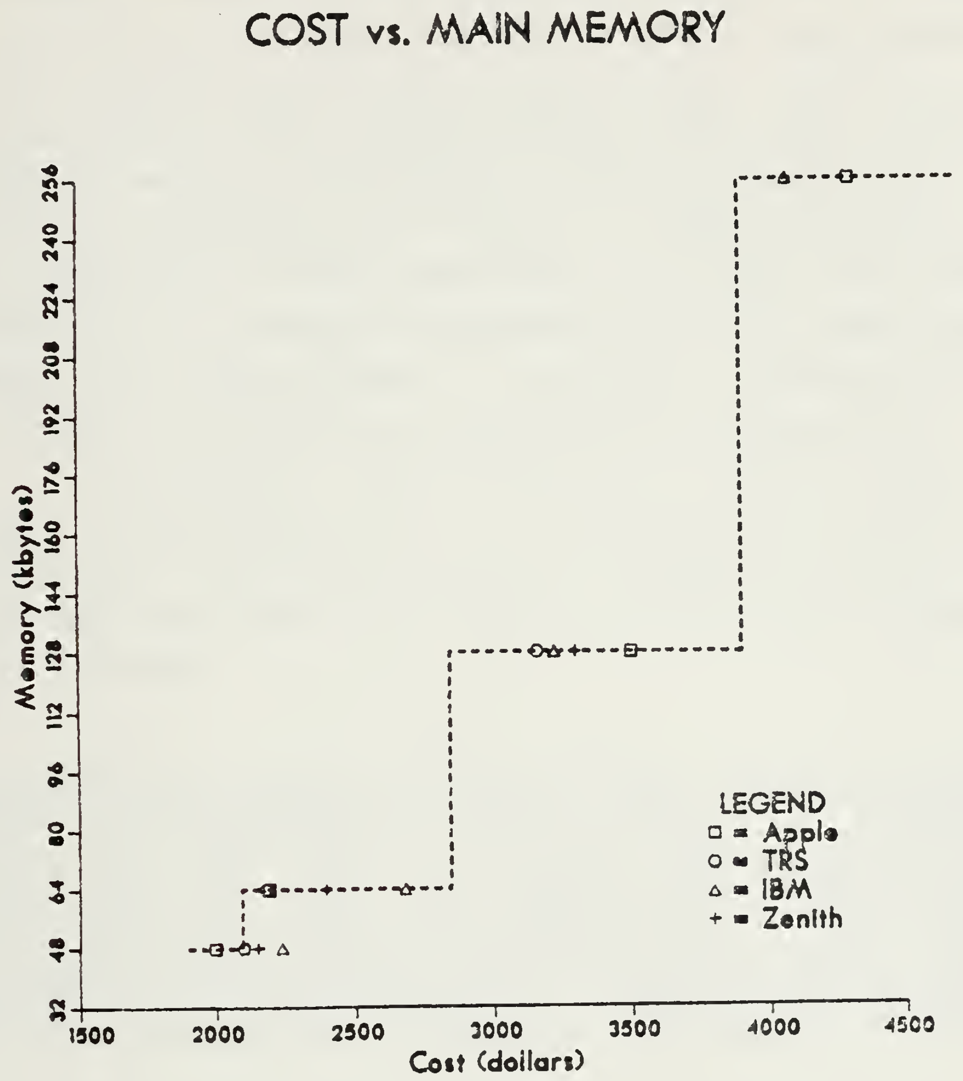
The hardware market is characterized by amazing advances in technology. The computer power of the 1960's Atlas mainframe can now be purchased for under $100. Manufacturers of microcomputers have generally upgraded their products in three basic ways. They have either: a) held price constant and increased performance; b) held performance constant and decreased price; or c) introduced a completely new architecture with brand new features. A current example of the latter is the introduction of the sixteen-bit microcomputer. These rapid advances in

technology impact upon lifecycle calculations of hardware cost. Also affecting lifecycle cost is the recent Accelerated Cost Recovery System (ACRS) method of equipment depreciation. Under ACRS, the cost of a microcomputer used in a business application can be fully depreciated over five years. This rapid write off provides a good degree of insurance against hardware selection mistakes.

Although not readily admitted by microcomputer manufacturers, their individual products are purchased by consumers without regard to the status attached to the brand name. Purchase price, not the prestige of say, IBM, is the primary consideration. For example, in late 1982 Apple Computers reduced the purchase price of the Apple II+ model by approximately 25 percent. Sharp sales declines prompted Radio Shack, Zenith, and IBM to make comparable reductions on their similar systems [Ref. 15: p. 456]. This indicates a high cross elasticity of demand. In fact, these systems were also experiencing sales pressure from Apple's forthcoming shift from MSI to LSI technology and market anticipation of 16 bit microcomputers. It can be assumed that this high cross elasticity means price is nearly independent of manufacturer, technology, and machine architecture. The primary element in hardware price is amount of main memory. Figure 3.5 illustrates costs for various memory configurations of four popular micros.

2. Costing Method

The cost of hardware (C_H) considered here will include two elements; the cost of the processor (designated C_c) and the cost of peripherals (C_P). Peripherals will include printers, disk drives, modems, and cards installed in the computer cabinet. Processor costs refer to only the CPU, co-processors, and memory. This differs from mainframe costing where the prices for these items are so variable



Source: Byte Magazine

Figure 3.6 Prices of Four Microcomputers.

they must be considered separately. As shown in Figure 3.6, hardware cost is primarily a function of memory size. Therefore, a unit of memory is a convenient measure when referring to the size of the microcomputer. The memory required is a function of the applications selected to run

on the computer. The memory must be large enough to contain the program and data. This requirement is becoming less and less binding as new operating systems which are capable of "swapping pages" of memory are introduced. Swapping is the technique whereby main memory contains only that amount of information needed to perform the current program step. The additional instructions and data are swapped back and forth from a secondary storage device such as a disk. The cost of peripherals can likewise be considered as a step function dependent upon applications. A printer or modem will only be purchased if required by the intended application.

At this point, what is needed is a way to cost hardware and peripherals as a non-recurring cost while accounting for the historical price reduction of the items. Ein-Dor and Jones [Ref. 15: p.6] provide a good discussion of this problem:

> It is assumed that the market price of hardware can be estimated as a function of time. This function (called the unit cost estimating relationship), is written u $(t,\bar{u})$. The inclusion of the time variable models the observation that hardware prices have changed most dramatically over time. In general, for a given technology, intuition (and casual empericism) suggests that market prices decline over time--second and third-hand sales are at lower prices than first time sales. Thus the time rate of change (designated du/dt) is generally negative for computer hardware. The variable $\bar{u}$ represents the parameters of the estimating function.

The parameters referred to, in conjunction with the variable $\bar{u}$, are the parameters that locate the price curve of the item. For example, if the price curve is linear, then equation 3.7 shows the equation for the price curve and $\bar{u}$ is

$$u = at + b \qquad \text{(eqn 3.7)}$$

used to estimate a and b. Equation 3.8 modifies equation 3.7 to indicate the dependence of the parameters a and b upon $\bar{u}$.

$$\mu = a(\bar{\mu})t + b(\bar{\mu}) \qquad \text{(eqn 3.8)}$$

When dealing with microcomputers, an additional variable applied to the unit cost estimating relationship will yield the desired total cost. That variable is the number of units of memory needed by the application that uses the most memory (designated H_c). Therefore the total cost of hardware at the time of start-up (designated T_B) is

$$C_c = H_c \mu_c(T_B, \bar{\mu}_c) \qquad \text{(eqn 3.9)}$$

equation 3.9 . Likewise, where H_p equals the maximum cost of peripherals over all applications, the cost of peripherals at start-up is given by equation 3.10 .

$$C_p = H_p \mu_p(T_B, \bar{\mu}_p) \qquad \text{(eqn 3.10)}$$

It should be pointed out that where H_c refers to the maximum of one application, H_p refers to the maximum over all applications. For example, H_p is the sum total of the cost of the printer for application one; the language card for application two; the disk drive for application three; and so on.

The salvage value of both hardware types can also be determined using the unit cost estimating relationship. In this case, the time at which production ceases (designated T_E) is used to locate the values. The equations for

$$S_c = H_c \mu_c(T_E, \bar{\mu}_c) \qquad \text{(eqn 3.11)}$$

$$S_p = H_p \mu_p(T_E, \bar{\mu}_p) \qquad \text{(eqn 3.12)}$$

salvage values are written in equations 3.11 and 3.12 . In these and all time-dependent equations, T_B and T_E are calculated based on the present as time zero.

Summing the two values defined by equations 3.11 and 3.12 will yield S_V , the total hardware salvage value. This value must be discounted to the present and subtracted from the costs to yield the net present value of the hardware. This is accomplished by applying the appropriate discount rate (r) as shown in equation 3.13 where C_H is the sum of

$$NPV_H = C_H e^{-rT_B} + S_V e^{-rT_E} \qquad \text{(eqn 3.13)}$$

C_c and C_p , the results of equations 3.9 and 3.10 . Selection of the discount rate is left to the individual.

3. Hardware Maintenance Cost

Since software maintenance is performed only by the software vendors as discussed above, the only maintenance costs in this scenario are hardware maintenance costs (C_m). These costs are spread over the entire lifecycle of the project and are the only recurring costs of automation. Materials such as diskettes and paper are overhead costs and will not be included in this analysis. The symbols p_c and p_p will be used to designate the price of maintenance of the processor and peripherals, respectively. At each instant of production, the maintenance costs are illustrated by equation 3.14, where the zero subscript means that the

$$C_{m_0}(t)\,dt = (H_c P_c + H_p P_p)\,dt \qquad \text{(eqn 3.14)} \cdot$$

variable represents any given instant in time. Therefore, the net present lifecyle maintenance costs ($NPLC_m$) can be derived by adding all the instantaneous values by using integration and discounting to the present as in equation

$$NPLC_m = \int_{T_B}^{T_E} C_{m_0}(t)\,dt = \int_{T_B}^{T_E} (H_c P_c + H_p P_p)\, e^{-rt}\,dt \qquad \text{(eqn 3.15)}$$

3.15 . This maintenance cost can also be called the operating cost of the system.

C. OTHER COSTS

1. Risks

The risk of failure of an automated project must be considered as a cost. It is not quantifiable since the probability of failure cannot be estimated with any accuracy. The risk of failure can be substantial as this quotation from Levy [Ref. 17: p. 204] illustrates:

> Small businessmen are usually more demanding than other customers because often their entire business operation depends on the computer. ... Neither the small computer business nor its small business customer can survive a major financial mistake.

Wickham [Ref. 10: p. 51] agrees:

> The risk to the potential first time small business computer user is a major deterrent to the use of computers in many cases. When the business owner considers the extent to which he is playing 'You Bet Your Company' on the successful transition to a new computer system, he becomes very cautious. The risk is inherent in the managers lack of experience and knowledge in the area of computers, but also due to the fact that the system size and cost is such that he must place as many applications as possible on the system in order to justify its cost. This of course, further increases the risk he is taking.

Even with a low probability of failure, the expected value can remain quite high since the entire business may depend upon the computer. Haftka [Ref. 3] relates the situation of a failed attempt by professional computer scientists to use

an eight-bit micro for a structural engineering application. Problems arose froma a totally unexpected source; the microcomputer's eight-bit representation of numbers resulted in poor data accuracy. The micro could not carry the significant digits needed for precise structural analyses.

A corollary to the risk of failure is the risk of using a poorly designed system, one that does not actually meet the needs of the individual. The expense involved is the opportunity cost of using a better, more effective method or perhaps improving performance by using the old manual system. Rather than admit failure and write off the experience as a loss, users will often force themselves to use a computerized system. Some examples of poor systems are:

1. Systems that require continuous data encoding for machine readability.
2. Systems producing cluttered output.
3. Systems with poor data accuracy.
4. Programs which force users through many steps when a simple update is all that is needed.
5. Systems which cannot be used without frequent referral to instruction manuals.

Since these costs cannot be determined, they will not be included in the decision rule.

2. Design and Start-up Costs

Design and start-up costs are extremely difficult to quantify. They represent the opportunity costs of the time lost while building and implementing the system. There is much in the literature dealing with mainframe and large system start-up costs, but these examples cannot be scaled down to apply to microcomputers. This is due to the fact that most of these costs are incurred by computer system professionals, not the system user, a relatively unskilled layman. One of the primary elements in this cost category

is the expense of converting flat paper files into a computer database. There are three basic methods of file conversion; mass conversion, conversion at time of account service, and conversion of new accounts. The latter two methods carry an additional burden of having two systems in operation simultaneously. For purposes of the decision rule, these costs will not be considered. If accurate estimates of these expenses can be obtained, they should be considered as a non-recurring cost encountered at system start-up time.

IV. DECISION RULE DEVELOPMENT

Now that the costs and benefits have been measured, development of a set of decision rules to permit optimization can begin. The previous chapter discussed optimal levels of hardware and software separately. This is useful for a general explanation of the optimum amounts but does not cover the specific combination in detail. The decision rules are based on the technique of linear programming to achieve optimization. The form of the linear program seeks to maximize or minimize an objective which is a function of one or more decision variables, subject to a set of constraints. The initial decisions on quantity and mix, timing, and allocation form constraints on the primary economic decision rule.

A. DECISIONS

1. Quantity and Mix

The quantity and mix decision attempts to optimize the amount of hardware and software purchased. Since the amount of software drives the decision on how much hardware to procure, the mix decision is trivial. However, the cost of additional hardware does affect the decision on how much software to purchase. Some marginal measure of value change in relation to a change in the quantity of software is needed to calculate the desired quantity of software. The section on revenue generation in Chapter II noted that the microcomputer does not actually contribute revenue to the user but produces benefits which have a measurable value. Assuming this value is created by the machine (with the applicable software), there is some production function,

Φ_V, which defines this value creation. The target measure can be expressed by the change in the production function with respect to the change in the quantity of the sole

$$MVP_s = \frac{d\Phi_V}{dq_s} \qquad \text{(eqn 4.1)}$$

input, software. Equation 4.1 is the resultant measure. It shall be called the marginal value product of software (MVP). This is at variance with the traditional definition of marginal value product which refers to the specific case of the marginal revenue product of a good. However, the term marginal value product will be used here to underscore the facts that: a) no physical product is produced (hence marginal physical product is inappropriate); b) the system generates no direct revenue (which means marginal revenue product is not the correct term); and c) the benefits produced have a measurable value. Therefore, marginal value product seems to be the most correct term for this unique case although it is not the textbook definition.

The marginal increase in software may require an additional amount of hardware and increased maintenance. So the net present value of the hardware must be considered along with the acquisition cost of the software which will form the opportunities foregone by the project. The acquisition cost of the software (C_S) from equation 3.4 is added to the net present value of the hardware from equation 3.13 and the lifecycle maintenance cost as figured in equation 3.15 . Therefore, the overall net opportunities foregone (which will form the numerator of the decision rule) are

$$C_s + C_H e^{-rT_B} - S_V e^{-rT_E} + \int_{T_B}^{T_E} H_c P_c + H_P P_P e^{-rt} dt \qquad \text{(eqn 4.2)}$$

found by equation 4.2. Now by applying the lifecycle values gained by software utilization to equations 4.1 and 4.2 the formal decision rule can be derived as equation 4.3. The marginal value product of equation 4.1 is divided by one to form the denominator of the right hand side of equation 4.3. Thus the right hand side is the net opportunities foregone divided by the marginal value product or, the net opportunities foregone at the margin. The left hand side of equation 4.3 is the lifecycle marginal value of the system as found by equation 3.2, where the quantity of the input, q, in equation 3.2 refers again, to the sole input, software. In words, the optimal quantity of software will be purchased, ceteris paribus, when the marginal present lifecycle value of the system equals the marginal present lifecycle cost of the system measured in terms of software.

$$\int_{T_B}^{T_E} NVe^{-rt} = \left(\frac{d\Phi_V}{dq_s}\right)^{-1}\left[C_S + C_K e^{-rT_B} + S_V e^{-rT_B} + \int_{T_B}^{T_E} H_c P_c + H_p P_p e^{-rt} dt\right] \qquad \text{(eqn 4.3)}$$

2. Timing

The decision on when to begin microcomputer operations is very important. As discussed in Chapter II, the price of hardware has decreased greatly over time. The timing decision rule accounts for the net opportunities gained and foregone at the margin by starting operations at time T_B and ceasing operations at time T_E. The marginal net opportunities gained at T_B is the value of the system less the operating (maintenance) costs. Equation 4.4 is derived from equation 3.14 and shows these opportunities discounted

$$(V - H_c P_c - H_p P_p)e^{-rT_B} \qquad \text{(eqn 4.4)}$$

to the present. Now equation 4.4 must be decreased by the cost of not delaying purchase in a market where the prices are falling. As mentioned earlier, Ein-Dor and Jones designated this price reduction as du/dt. Adapting this notation and substituting the relevant variables yields equation 4.5, the representation of the costs of purchasing hardware at

$$\left(\frac{du_c}{dT_B} H_c + \frac{du_p}{dT_B} H_p\right) e^{-rT_B} \qquad \text{(eqn 4.5)}$$

time T_B. Since prices are falling, equation 4.5 should yield a negative value. The next procedure is to add equation 4.4 and 4.5 to form equation 4.6, the total

$$\left(V - H_c p_c - H_p p_p + \frac{du_c}{dT_B} H_c + \frac{du_p}{dT_B} H_p\right) e^{-rT_B} \qquad \text{(eqn 4.6)}$$

opportunities gained at time T_B. Now that the opportunities gained have been adjusted for the opportunity cost of the purchase time, equation 4.6 can be called the time marginal net value plus the time marginal acquisition expenditure adjusted for opportunity cost. Note that here again the values of the hardware variables refer to that amount of hardware needed to implement the applications for which software is obtained.

The opportunities foregone at the margin are simply what could be done elsewhere with the funds invested in the system. This figure is obtained by multiplying system cost from equations 3.9 and 3.10 times the appropriate rate of return. Therefore, equation 4.7 represents the net opportunities foregone due to system start-up. Equation 4.7 is also called the time marginal interest value on acquisition expenditure opportunity cost. Since the optimum is defined as the point where the net marginal value is zero, setting the marginal opportunities gained (marginal value) equal to

$$r\left(C_s + H_c \mu_c\left(T_B, \bar{\mu}_c\right) + H_p \mu_p\left(T_B, \bar{\mu}_p\right)\right) e^{-rT_B} \qquad \text{(eqn 4.7)}$$

the marginal opportunities foregone (marginal cost) will produce the desired decision rule. Equation 4.8 is formed

$$\left(V - H_c P_c - H_p P_p + \frac{d\mu_c}{dT_B} H_c + \frac{d\mu_p}{dT_B} H_p\right) e^{-rT_B} =$$

$$r\left(C_s + H_c \mu_c\left(T_B, \bar{\mu}_c\right) + H_p \mu_p\left(T_B, \bar{\mu}_p\right)\right) e^{-rT_B} \qquad \text{(eqn 4.8)}$$

by setting equation 4.6 equal to equation 4.7 . The discounting factors on either side of the equality sign cancel out. This means the results need not be expressed in now year dollars. The decision rule can now be stated as: the optimum time to begin system use, *ceteris paribus*, is when the time marginal net value plus the time marginal acquisition expenditure adjusted for opportunity cost equals the time marginal interest value on acquisition expenditures opportunity cost.

The decision rule on when to cease operations is computed in exactly the same way except now the opportunities are gained and foregone by continuing operations. The system end time (T_E) is used in place of system start time (T_B). Also the cost of software realized at start time now appears as a charge against system value. This assumes software cannot be sold with the system. Equation 4.9 is

$$\left(V - C_s - H_c P_c - H_p P_p + \frac{d\mu_c}{dT_E} H_c + \frac{d\mu_p}{dT_E} H_p\right) e^{-rT_E} =$$

$$r\left(H_c \mu_c\left(T_E, \bar{\mu}_c\right) + H_p \mu_p\left(T_E, \bar{\mu}_p\right)\right) e^{-rT_E} \qquad \text{(eqn 4.9)}$$

the result of these changes. Equation 4.9 states that the optimum time to cease system operation, *ceteris paribus*, occurs when the time marginal system value plus the time

marginal hardware salvage value adjusted for opportunity cost equals the foregone time marginal interest value of the hardware.

3. Allocation

The allocation decision rule measures opportunities gained and foregone at the margin by using the computer system. In this discussion the terms value and revenue have some important implications. Recall that microcomputer use in and of itself does not generate revenue but rather saves cost and time. The time savings has a value in that the extra time can enable the user to generate revenue for the firm. Therefore, the opportunities gained by using the system will be called value. If the user is not using the system, it is assumed he will be engaged in some activity that will produce revenue for the firm. Hence the opportunities foregone by the user using the system are the revenues he could have generated if he were not using the microcomputer. The term revenue will be used when speaking of the opportunities an individual can gain without using the micro. As will be shown later, plots of revenue and value over time form different curves. It is the nature of these differences that permit calculation of the optimum time allocations. A brief discussion of revenue generation in a small firm is required at this point to amplify the distinctions between value and revenue.

Unlike large computing systems serving many users, personal computers are not in continuous use. They are designed to serve a single user and, thus, are used in much the same way as a telephone to accomplish a specific task. Therefore, the user should develop some insight on how much time to use the system and how to allocate that time among the various applications. This, of course, is highly dependent upon the nature of the market affecting the user.

Consider the case of a small business operating in a perfectly competitive market. This is a valid assumption, for although an industry may be dominated by a few large firms, the market segment for the remainder of suppliers may operate under perfect competition. Mansfield [Ref. 18: p. 127] states that under perfect competition a firm will possess demand and marginal revenue curves that are horizontal and equal. The line labelled MR in the lower graph of Figure 4.1 shows Mansfield's marginal revenue curve over time. Therefore, assuming the potential computer user dedicates more time to revenue generation, the best that can be expected is a monotonic increase in total revenue over time.

This linear increase in total revenue (TR) over time (t) means that the time marginal revenue or marginal opportunities foregone by system use represented in equation 4.10 reduce to a constant (α). The time marginal opportunities (value) gained by using the system, however, have a point of diminishing marginal returns as shown by Figure 4.1 . The marginal value curve in Figure 4.1 has an equation as shown in equation 4.11 . Naturally, the user will perform the function that has the largest marginal contribution to his firm. There is some time, shown on Figure 4.1 as t*, when the marginal revenue of non-use are greater than the marginal value of using the system. At this point the user will stop working with the computer and begin revenue generating activities. The decision rule is then, the optimum amount of system use is achieved, _ceteris paribus_, when the time marginal revenues gained from not using the system equal the time marginal values of system use. It is expressed by equation 4.12 where use time, t, is less than T, the total time the system is available.

Now that the optimum amount of system use has been identified, the user must try to maximize the total and marginal values of the system by choice of applications.

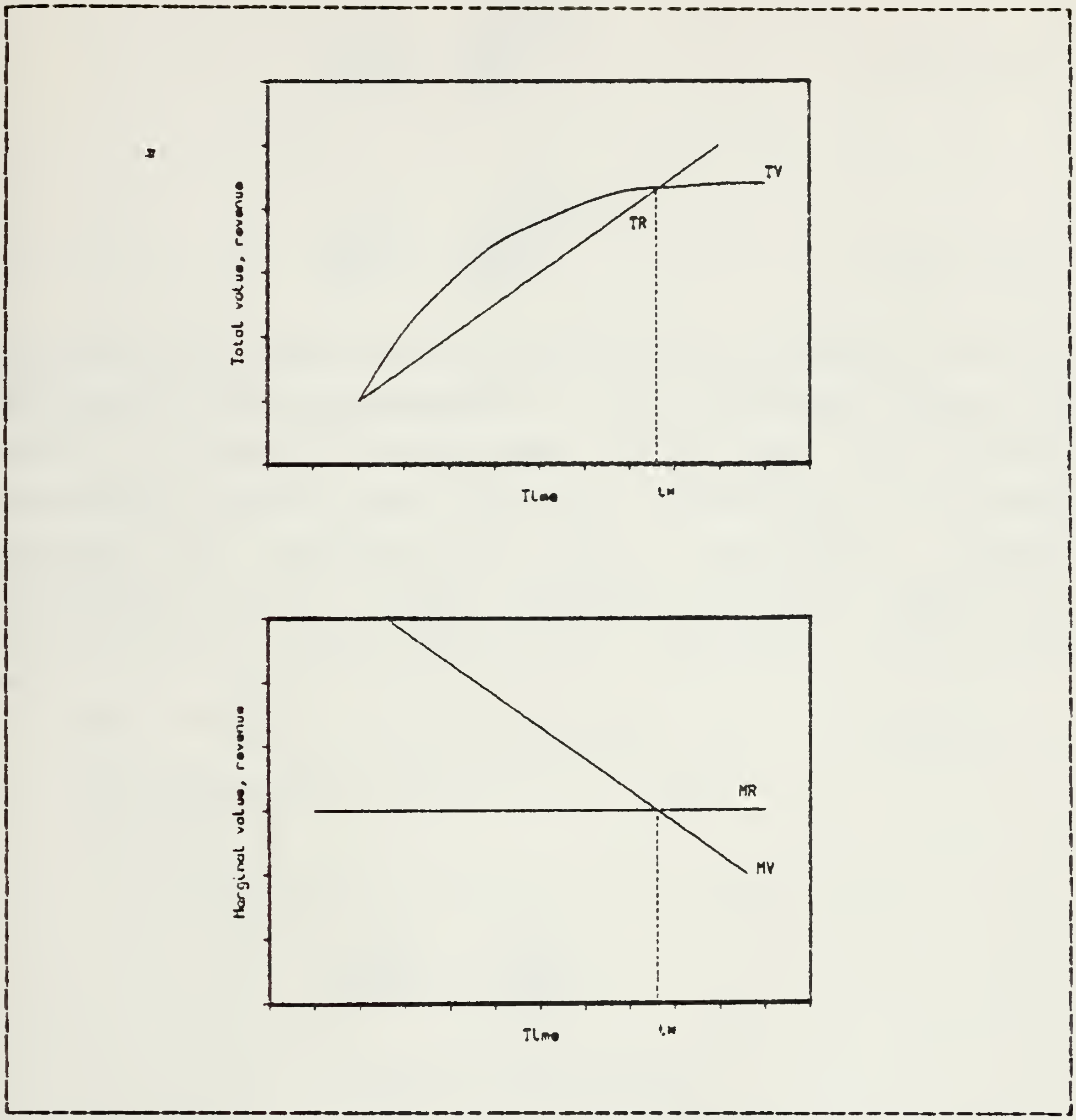


Figure 4.1 Value-Revenue Relationships in a Small Firm.

$$\frac{dR}{dt} = \alpha = MR \qquad \text{(eqn 4.10)}$$

Referring again to Figure 4.1, the user will attempt to increase the y axis values of the TV and MV curves by

$$MV = \frac{d \sum_{i=1}^{N} v_i}{dt} \qquad \text{(eqn 4.11)}$$

$$\frac{dR}{dt} = \frac{d \sum_{i=1}^{N} v_i}{dt} \qquad \text{(eqn 4.12)}$$

selecting the amount of time the various programs are to be executed on the microcomputer. This procedure enables the user to attain the maximum value from the system. Maximizing system value can be expressed as an integer program with equation 4.12 as a constraint. Integer programming is the proper technique since an application program has no value unless it is completed. The program is:

$$\text{Max } Z = x_1/t_1 \; v_1 + x_2/t_2 \; v_2 + x_3/t_3 \; v_3 + . \; . \; . \; x_N/t_N \; v_N$$

By choice of: $x_1, x_2, x_3, . . . x_N$.

Subject to:

$$x_1/t_1 + x_2/t_2 + x_3/t_3 + . \; . \; . \; x_N/t_N = t^*$$

$$\frac{dR}{dt} = \frac{d \sum v_i}{dt}$$

$$t < T$$

and non-negativity.

In the integer program, the x values are the decision variables which represent the optimum amount of time to run each application (the subscripted v variables). The subscripted t variables represent the run time of each application. The quantities x/t are then the amount of run time divided by the required run time to yield the number of times each application should be run in time t*, the optimum amount of system use time. Equation 4.12 appears here as a constraint

as a reminder that the optimum amount of system use time has already been determined and cannot be violated. Notice that hardware costs play no part in the allocation decisions. Since the investment has already been made, these costs are known as sunk costs and are not relevant to the decision.

B. SYSTEM DECISION RULE

The system decision rule attempts to maximize system value by selection of both hardware and software. It is expressed as a linear program encompassing many of the equations previously discussed. The rule calculates the maximum value that can be achieved by the system under optimum conditions and provides the basic figures for lifecycle cashflow analysis. The system decision rule is:

$$\max Z = \int_{T_B}^{T_E} V e^{-rt}\,dt - C_S - C_H + S_V - \int_{T_B}^{T_E} C_m e^{-rt}\,dt$$

By choice of: C_S, C_H, T_B, T_E

Subject to:

$$V = \sum_{i=1}^{N} v_i \qquad \text{eqn } 3.5$$

$$C_S = \sum_{i=1}^{N} c_i \qquad \text{eqn } 3.4$$

$$C_H = H_c u_c(T_B, \bar{u}_c) + H_p u_p(T_B, \bar{u}_p)\, e^{-rT_B} \qquad \text{eqn } 3.9,\ 3.10$$

$$S_V = H_c u_c(T_E, \bar{u}_c) + H_p u_p(T_E, \bar{u}_p)\, e^{-rT_E} \qquad \text{eqn } 3.11,\ 3.12$$

$$C_m = H_c p_c + H_p p_p \qquad \text{eqn } 3.14$$

$$T_B < T_E$$

and non-negativity.

Notice that this is not a marginal computation, but a straightforward accounting of system value. All factors are discounted to the present where appropriate.

C. SENSIVITY ANALYSIS

Sensitivity analysis identifies those variables within the decision rule which introduce the greatest amount of change in the final outcome. Two variables which introduce great fluctuations in the equation are interest rate (r) and lifecycle (t). These serve as magnifiers of the recurring costs and can introduce wide variations in the results. The lifecycle figure can also have great impact if workload growth over the lifecycle exceeds the amount allotted for in hardware size. The rule should be recalculated with several sets of lifecycle and rate values while keeping the other decision variables constant. This will help identify possible alternative solutions.

The quantity of software produced is another variable which has a large effect on the decision rule. Since hardware is selected to fit the software, the quantity of software determines both processor and peripheral costs as well. These costs in turn are used to figure hardware maintenance cost, a recurring cost throughout the lifecycle. This means that the initial selection of software is critical not only in the value determination previously discussed, but as a cost factor that is the basis of many other computations. In addition, most of these costs are incurred at the system start time and are not greatly affected by discounting. This "front loading" of costs in now year dollars means a substantial commitment is at risk in the project.

D. ECONOMIES OF SCALE

A final topic in this economic analysis concerns economies of scale. Assuming the user is about to procure a microcomputer, the question of whether greater savings can be achieved by moving to a larger scale computer is germaine

to this discussion. Since the micro represents the smallest scale computer, the existance of economies of scale should dictate against a small computer purchase. The issue of economies of scale was first postulated by Herbert Grosch in the 1940's. Grosch's law states that computer equipment average cost decreases substantially as size increases. Sharpe [Ref. 9: p.315] interpreted Grosch's law in the form of equation 4.13 where C is system cost; E is effectiveness

$$C = K\sqrt{E} \qquad \text{(eqn 4.13)}$$

(performance, speed, throughput); and K is some constant. Subsequent findings by Solomon [Ref. 19] and Knight [Ref. 20] indicate Grosch's law applies more to scientific computing and other CPU intensive processing than business applications but is generally true over a wide range of computing uses. Does this mean that diseconomies are automatically introduced by the very nature of small computers? Hardly; these findings are all based on large operations where there is a continuous job stream. The personal user has perhaps a dozen applications, not nearly enough to keep a mainframe busy. Economies of scale do exist, but only when the amount of work is sufficient to warrant the large computer.

V. CONCLUSIONS

This paper is an initial attempt to clarify the benefits gained by microcomputer use. There is a surprising lack of literature on the economics of small systems. Most of these equations are based on information gained by studying large systems and applying it to the small system market. There is a great deal of literature offering general, non-parametric advice to the personal computer user. Unfortunately, most authors prefer to discuss hardware and overlook the real value-producing object, software. This hardware orientation results in situations like that of Standard Oil of Indiana. Standard spent a great deal of time and money selecting the best microcomputer for use by their executives. The culmination of this effort was a lengthy report on microcomputers [Ref. 21] and inclusion of the approved systems in their qualified products list [Ref. 22]. This means that executives are authorized to purchase any of the qualified systems with no guidance as to effective software. In terms of the decision rules discussed above, this is clearly a mistake.

This paper presents the hypothesis that production of value using a microcomputer involves only one input, software. Hardware is obtained only as a device to accomplish the software's work and its size is dependent upon the requirements of the software. In this setting, hardware is analogous to a catalyst in chemistry; the presence of a catalyst is required for a reaction but is not actually an input. The decision rules reflect this assumption by relating the costs of hardware to the amount of software required. Since memory has become increasingly less expensive and microprocessor speeds increased, programmers have

become less concerned about limiting program size. The hardware costing decision rules show that a savings can be gained by keeping memory requirements low.

More empirical research is needed to help determine how value is produced by a microcomputer. If the production functions of various microcomputers can be expressed econometrically, it would reduce the complexity of some of the decision rules and provide some hard numbers to work with. Also, the effect of various pieces of software on the economic decisions is not clearly understood. More basic research into the nature of information quality is needed. For example, what are the opportunity costs gained and foregone when selecting an accounting software package from vendor A instead of vendor B? The answer depends upon the valuation of the effectiveness of the two competing software packages. At this time no empirical method exists for measuring effectiveness of quantities of software.

Consideration of the recreational value of microcomputers has been purposely excluded from the decision rule since the hobby application does not contribute to the firm's revenue. This may not be a valid assumption. Evidence exists that computer power in the hands of the individual has some value to the user as a perquisite of his position. In fact, in a Business Week article on marketing small computers [Ref. 23: p. 78], Warren Winger, chairman and owner of the Compu Shop chain observes,

> Personal computers are very much like single engine business aircraft. They're bought for business, but most of the activity at smaller airports is on the weekend.

If there is no demonstrable value gained by such recreational use, there is at best some psychological benefits or image enhancement to be gained by the user by having a computer system at his disposal.

The importance of systems analysis cannot be overemphasized. This paper has suggested a systematic method for analyzing benefit elements designed to tailor the microcomputer to the individual. To achieve the maximum benefit it is essential that the applications be clearly identified. This method of benefit analysis attempts to combat the unknown quantities mentioned above in an informal, i. e. non-mathematical, way, albeit within a well-specified framework. Most of the variables used in the decision rules are based on the selection of effective software to meet the individuals' needs. All the software in the world will be of no value if the user does not understand the nature of his work and how to accomplish it.

APPENDIX A

VARIABLE DIRECTORY

V = value obtained.

TV = total value.

TC = total cost.

MV = marginal value.

MC = marginal cost.

NV = net value.

R = revenue.

TR = total revenue.

MR = marginal revenue.

V_s = value obtained from software.

c_i = cost of one application.

v_i = value of one application.

q_s = quantity of software.

C_s = cost of software.

C_H = cost of hardware.

C_c = cost of processors.

C_p = cost of peripherals.

u_c = a unit of processor.

u_p = a unit of peripheral.

H_c = total amount of processor required by software.

H_p = total amount of peripherals required by software.

S_c = salvage value of processors.

S_p = salvage value of peripherals.

S_V = total salvage value.

C_m = total cost of maintenance.

t = lifecycle.

T = amount of time in lifecycle (t) that the system can be operated accounting for maintenance down time and user time off.

T_E = time of system cessation.

T_B = time of system start-up.

t_i = time required to complete application i.

r = discount or interest rate.

ϕ_v = microcomputer production function.

p_s = price of software.

p_c = price of processor maintenance.

p_p = price of peripheral maintenance.

LIST OF REFERENCES

1. Libes, S., "Bytelines", Byte, V. 7, n. 8, August 1982.

2. Emery, J. Cost/Benefit Analysis of Information Systems, The Society for Management Information Systems, 1971.

3. Office of Naval Research Report ONR N00014-80-C-0364, The Cost Effectiveness of Minicomputers vs. Mainframes for Structural Analysis Problems, by Haftka, R. T. and von Mayrhauser A. K., 31 August 1981.

4. Gupton Jr., J. A., Getting Down to Business with Your Microcomputer, Sourcebooks, 1979.

5. Barden Jr., W. T. How to Buy and Use Microcomputers and Minicomputers, Howard W. Sams and Sons, 1976.

6. Lu, Cary, "Microcomputers: The Second Wave", High Technology, V. 2, n. 5, September/October 1982.

7. Nahil, J. C., "Small Business Computers: A Beginner's Guide", Computers and People, V. 27, n. 2, February 1978.

8. Infotech, Computing Economics, Infotech State of the Art Report 12, Infotech Information Inc., 1973.

9. Sharpe, W. F. The Economics of Computers, Columbia University Press, 1969.

10. Wickham, R. F., The Impact of Microcomputers on Small Business Data Processing, paper presented at IEEE Computer Society International Conference, 16th, San Francisco, California, 28 February-3 March 1978.

11. Joint Tactical Communications Office, "Cost Effectiveness Program for Joint Tactical Communications (TRITAC)", V. 3, October, 1976, Joint Tactical Communications Office, Fort Monmouth, New Jersey.

12. Muller, D., "Personal Computers in Home and Business Applications", Computers and People, V. 26, n. 12, p. 12, December 1977.

13. Naval Postgraduate School Report number NPS55Js9061A, A Model of Administrative Choice in the Systems Analysis Environment, by C. R. Jones, 10 June 1969.

14. Lewis, T., "Missing Computer Software", Business Week, n. 2652, September 1, 1980, pp. 46-56.

15. Libes, S., "Bytelines", Byte, V. 7, n. 10, October 1982.

16. Ein-Dor, P. and Jones, C. R., Economics and Management of Computerized Information Systems, draft copy, 1982.

17. Levy, W. A., "Selling to the Small Business Market", Mini Micro Systems, V. 13, n. 5, pp. 203-208, May 1980.

18. Mansfield, E., Microeconomics: Theory and Applications, Norton, 1979.

19. Solomon, M. B., "Economies of Scale and the IBM System/360", Communications of the ACM, V. 9, n. 6, pp. 435-440, June 1966.

20. Knight, K. E., "Changes in Computer Performance", Datamation, V. 12, n. 9, pp. 40-54, September 1966.

21. Standard Oil Company (Indiana) Report U-2150, Evaluation of Personal Computers, by T. T. Cook, pp. 1-3, February, 1981.

22. Standard Oil Company (Indiana) Letter Serial 1006 to distribution list, Subject: Computing Devices Approved Hardware List, 19 July 1982.

23. Anonymous, "Computer Stores", Business Week, n. 2707, pp. 76-82, 28 September 1981.

BIBLIOGRAPHY

Batt, R., "Apple Targets Large Business in Market Push", Computerworld, V. 15, n. 41, p. 105, October 12, 1981.

Beilner, H. and Gelenbe, E., eds., Modelling and Performance Evaluation of Computer Systems, North-Holland, 1977.

Cotton, I. W., "Microeconomics and the Market for Computer Services", Computing Surveys, V.7, n. 2, pp. 95-111, June 1975.

Dade, T. P., An Economic Decision Model for Engineers, Master's thesis, Pennsylvania State University, 1973.

Datapro Research Corporation report SC20-100-101, Personal Computers, April 1982.

Gandolfo, G., Quantitative Analysis and Economic Estimation of Continuous Time Dynamic Models, North-Holland, 1981.

Genovese, D. H., An Economic Analysis of Life Cycle Military Manpower Maintenance and Training Requirements in Avionics Minicomputer and Microcomputer Systems, Master's thesis, Naval Postgraduate School, 1973.

Goke, L. R. and Warren , W. L., Linear-Cost Banyan Interconnection Networks for Multiple Microprocessor Systems, paper presented at Rocky Mountain Symposium on Microcomputers, 3rd, Pingree Park, Colorado, 19-22 August 1979.

Howell, J. F. and Teichroew, D., Mathematical Analysis for Business Decisions, Richard D. Irwin, 1971.

Jones, M. R. and Pooch, U. W. Evolution and Design of Microcomputers, paper presented at Symposium on Trends and Applications 1976: Micro and Mini Systems, Gaithersburg, Maryland, 27 May 1976.

Littrell, R. F., "Economies of Scale in the IBM 360 and 370", Datamation, V. 20, n. 3, pp. 83-88, March 1979.

Mc Burney, N. R., "The Personal Computer as an Interface to a Corporate Management Information System", Byte, V. 7, n.10, p. 315, October 1982.

Mc Farlan, F. W., Nolan, R. L., and Norton, D. P., Information Systems Administration, Holt, Rinehart, and Winston, 1973.

Phister Jr., M., Data Processing Technology and Economics, 2nd. ed., Santa Monica Publishing Company and Digital Press, 1979.

Rao, G. V., Microprocessors and Microcomputer Systems, Van Nostrand Reinhold, 1978.

Smith, Jr., D. L., *Method to Evaluate Microcomputers for Non-Tactical Shipboard Use*, Master's thesis, Naval Postgraduate School, 1979.

Solomon, M. B., "Economies of Scale and Computer Personnel", *Datamation*, V. 16, n. 3, pp. 107-110, March 1970.

Stibbens, S., "Personal Computers Come Out of the Closet", *Infosystems*, V. 28, n. 9, pp. 86-90, September 1981.

INITIAL DISTRIBUTION LIST

No. Copies

1. Defense Technical Information Center 2
Cameron Station
Alexandria, Virginia 22314

2. Library, Code 0142 2
Naval Postgraduate School
Monterey, California 93940

3. Department Chairman, Code 54 1
Department of Administrative Sciences
Naval Postgraduate School
Monterey, California 93940

4. Curricular Officer, Code 37 1
Computer Technology Programs
Naval Postgraduate School
Monterey, California 93940

5. Assistant Professor D. C. Boger, Code 54Bk 1
Department of Administrative Sciences
Naval Postgraduate School
Monterey, California 93940

6. Associate Professor N. R. Lyons, Code 54Lb 1
Department of Administrative Sciences
Naval Postgraduate School
Monterey, California 93940

7. Mr. Peter R. Lynn, MC 0804 1
Standard Oil Company (Indiana)
Post Office Box 5910-A
Chicago, Illinois 60680

8. LT Mark Doernhoefer, USN 2
Wargaming Division
Naval War College
Newport, Rhode Island 02840

Thesis 200637
D6323 Doernhoefer
c.1 Microcomputer microeconomics.