ASPECTS OF MICROCOMPUTER SOFTWARE DISTRIBUTION

Alan Henry Rostowsky

A dissertation submitted to the Faculty of Engineering, University of the Witwatersrand, Johannesburg, in partial fulfillment of the requirements for the degree of Master of Science in Engineering.

Johannesburg, 1985.

DECLARATION

I declare that this dissertation is my own, unaided work. It is being submitted for the degree of Master of Science in Engineering in the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination in any other University.

31st January 1985.

ABSTRACT

Since the microcomputer software industry began in 1975, its growth has been increasing exponentially each year. The purpose of this dissertation is to:

a) investigate the development of the early pioneers of the industry.

b) examine trends within the industry since its inception.

c) examine alternative methods of distribution of software.

d) to suggest an evaluation technique for Software Packages.

Most sources of information pertaining to the history and development of the industry were found in recently published international and local computer journals, technical publications and business journals.

The ev luation technique proposed by the author has been designed specifically for the retail sector of the distribution cycle; it may, however, be equally well utilised as a general guide for any individual investigating a microcomputer software product. To the author's knowledge, no similar structured evaluation technique exists in published form.

To Hazei

To Hazel

CONTENTS

Page

DECLARATION i

ABSTRACT ii

CONTENTS iv

LIST OF FIGURES vii

1 INTRODUCTION 1

1.1 Background to the industry 1

1.2 Intention 3

1.3 Reference material 4

1.4 Assumptions 4

2 THE EARLY PIONEERS 5

2.1 Introduction 5

2.2 Microsoft 6

2.3 Digital Research 8

2.4 Visicorp 9

2.5 Software Arts 14

2.6 Lotus Development Corporation 16

2.7 Paul Lutus 17

2.8 S ftware Publishing Corp 18

3 TRENDS WITHIN THE INDUSTRY 19

3.1 Introduction 19

3.2 Early development 19

3.3 The industry matures: 1980-1983 20

3.4 Consolidation: 1984 - 1985 21

4 DISTRIBUTION METHODS 25

4.1 Introduction 25

4.2 Electronic Distribution 25

4.3 Vending Machines 27

4.4 Vertical Packages 28

4.5 Broadcasting 29

4.6 Public Domain Software 29

4.7 Freeware 30

5 SOFTWARE EVALUATION TECHNIQUE 32

5.1 Introduction 32

5.2 Methodology 33

5.3 STEP 1 - Generic Classification 34

5.4 STEP 2 - Application Classification 42

5.5 STEP 3 - Package Evaluation 46

5.6 STEP 4 - Package Positioning 53

5.7 STEP 5 - Support Systems 58

APPENDIX A FUNCTIONAL SPECIFICATIONS FOR: 60

WORDPROCESSING 61

SPREADSHEET 67

DATABASE 73

REFERENCES 79

LIST OF FIGURES

Figure Page

1 Selection Process 76

2 Generic Classification Table 77

3 Support Systems 78

1. INTRODUCTION

1.1 Background to the Microcomputer Software Industry

"The software industry, the essential but sometimes overlooked child of the personal computer revolution, is growing up fast. Sales are increasing at a rate of 50% per year. New companies are springing up every week. Big corporations are starting to move into the market. New programs are being launched with heavy promotion and large ádvertising budgets. Software-only specialty stores are multiplying. In short a business explosion is occuring."(1)

The micro computer software industry started as the offshoot to the micro computer industry which first began in January 1975 when Ed Roberts, a former airforce captain, marketed his 'MITS Altair 8800' in kit form, by mail order through an electronics magazine (2). The Altair appeared on the front cover of 'Popular Electronics' in December 1974, priced at $375 in its stripped down form (3). In 1977, Commodore, which was then a leading manufacturer of hand-held calculators, produced the Commodore 'PET' or 'Personal Electronic Transactor'. It was in the same year that Steven P. Jobs and Stephen G. WozniaK produced and sold their first 'Apple' computer for $700 (4).

Since the industry began in 1975, software sales have increased exponentially every year with tne growth of the micro computer. Sales were $600m in 1981, $1bn in 1982, $1.5bn in 1983 with projected sales of 12bn in 1990, making the software industry almost equal in size to today's household-appliance industry. Industry experts estimate that for every $1000 consumers invest in computer haroware, they purchase an additional $300 worth of software during the first year after buying the machine (5).

There is no typical software author; schoolchild and business executive alike have been successful at writing programs. It may be suggested that the software author be seen as a "technical novelist" and, as the incidence of computer literacy increases in the world, so will the number of software programs. It is not known how many programs actually exist; estimates were from 8000 to 40,000 in 1984 (6). A mini industry has been established simply to keep track of the latest software titles! For example, the publisher of the 'Whole Earth Catalogue', Steward Brand, has produced the 'Whole Earth Software Catalogue.' A magazine called 'Billboard' charts the progress of top selling software packages in a similar manner to a "Top Twenty Records" hit parade (7).

The first primitive form of software appeared in 1834 when Charles Babbage, a mathematics professor at Cambridge University, designed a machine called the analytical engine to solve mathematical equations, utilizing punched cards as input. In 1951, the first commercial computer, 'UNIVAC I', was used by the

U.S. Census Bureau. The first widely accepted computer language was 'FORTRAN' (FORmula TRANslation) which was developed by a team at IBM in 1956. Other popular languages are 'BASIC', 'Pascal', 'Ada' and 'COBOL'. Two professors from Dartmouth University wrote 'BASIC' (Beginners All-purpose Symbolic Code) and 'Pascal' was named after the 17th century mathematician. 'Ada' has been developed as the official language of the US Department of Defence, while Grace Hopper created 'COBOL' (COmmon Business Oriented Language) in the nineteen forties (8).

1.2 Intention

The intention of this dissertation is to:

a) investigate the development of the early pioneers of the Software Industry, with particular emphasis on Digital Reasearch, Microsoft and VisiCorp.

b) examine trends within the Software Industry since its inception.

c) examine alternative methods of distribution of software.

d) to suggest an evaluation technique for Software Packages.

1.3 Reference Material

Most sources of information were found in recently published international and local computer journals, technical publications and business magazines. Due to the fact that the software industry has really developed only in the last three years, no comprehensive subject matter on the industry's history and development could be found by the author. Currently available literature pertaining to evaluation techniques of packaged software is limited to application feature checklists comparing different software packages; to the author's knowledge, no structured evaluation technique exists in published form.

1.4 Assumptions

It is assumed that the reader is familiar with the technical aspects of the personal computer and associated software packages.

2. THE EARLY PIONEERS

2.1 Introduction

In the late 1970's, when the personal computer industry was in its infancy, a number of small software companies were started. "Many firms were started by one or two people with one computer, a good idea and some knowledge of computer programming." (9). Indeed, it is evident that those early entrepreneurs, working in backrooms and college dormitories, were the pioneers who laid the very foundations of the industry.

By early 1977, the basic "building blocks" of the industry were taking shape. Microsoft 'BASIC' broadened the base of programmers who could work with personal computers, while CP/M promised a large standardized market for those programs. The third element, the successful application package, developed in 1979 with 'VisiCalc' (10).

Currently, there are an estimated 1000 small to medium size software publishing firms, yet only a fraction are performing well. Among those exhibiting "healthy sales volumes, a competent sales force, a number of poular programs and money in the bank are Ashton Tate, Software Publishing, Microsoft, Digital Research and Lotus Development Corporation " (11).

It is the author's intention to examine the development of the early pioneers of the industry, Microsoft and Digital Research. A more detailed examination will be undertaken on VisiCorp and Software Arts, the two companies who developed 'VisiCalc'. Mention will be made of Paul Lutus, Lotus Development Corporation and Software Publishing.

2.2 Microsoft

In 1975, Bill Gates, then a 19 year old a law student at Harvard, purchased an 'Altair computer' produced by MITS which had been featured on the cover of 'Popular Electronics'. Together with a friend, Paul Allen - then 22 years old - who worked for Honeywell Inc., he began writing in his college dormitory what is currently known as 'Microsoft BASIC' (12).

Allen joined MITS in Albuquerque as director of software and persuaded Gates to join as well. The two set up Microsoft Inc. while working at MITS to market their 'BASIC' language.

Microsoft developed in a suburb of Seattle, Washington and grew as a company specializing in programming languages.

When IBM first developed the PC and required advice and design suggestions during 1981/82, Gates sold them his 'Microsoft Basic', which by then had become an industry standard. When IBM required an operating system for the PC, he suggested CP/M, but after Digital Research failed to reach an aggreement with IBM, Gates bought in an alternative from a small hardware company in Seattle, - Seattle Computer Products - and called it 'Microsoft Disk Operating System' or 'MS-DOS'. IBM adapted 'MS-DOS' for the PC, calling it 'PC-DOS'. As a result, Microsoft has cultivated a very close relationship with IBM and this has given the company a distinct advantage in setting hardware and software standards.

While Microsoft has derived obvious benefits from its association with IBM, it may be argued that the company's real strength is that it has been successful with more than one program and even more than one operating system. In addition to 'MS-DOS', Microsoft also sells 'Xenix', a version of 'Unix'.

In 1982, Gates and Microsoft assisted in the design of the successful 'Radio Shack Model 100' computer. Microsoft also received one of the first prototypes of the 'Apple Macintosh' computer to develop software. 'Microsoft Basic' was one of the first packages available on the "Mac".

Microsoft was responsible for the development of 'MSX', a new operating system that will enforce a standard in the area of home computers, and on which the new range of Japanese home computers is based.

In 1978, Microsoft had 15 employees. By April 1984 the company had grown to a staff of 510, selling 29 products including operating systems, languages, application systems as well as add-on boards and peripherals. In 1980, revenues were $4m. In 1984, sales were expected to reach $100m (13).

2.3 Digital Research

In 1975, Gary Kildall, a consultant at Intel, was writing a language called 'PL/M' for Intel's development systems. Together with a friend, John Torode, Kildall developed the basis of an operating system for the newly developed floppy disk and called it 'CP/M', short for 'Control Program for Microcomputers' (14).

The program was offered to Intel who turned it down and Kildall failed to impress any of the then dominant microcomputer manufacturers with his system. After Glenn Ewing of Imsai approached Kildall for a licence, the idea was developed to consolidate all of the hardware dependent portions of 'CP/M' into one section, so that anyone could buy a copy of 'CP/M' and perform the necessary modifications to ensure compatibility. With this change, the rapid proliferation of 'CP/M' through the industry began. Because it allowed users to switch easily from one brand of machine to another, 'CP/M' became the world standard for operating systems in the early part of the decade.

Together with Dorothy McEwen, now his wife, Kildall formed

Digital Research Inc. (DRI) in his toolshed in Pacific Grove, California.

In 1981, DRI had revenues of $6m. Forecasts for 1982 were $20m, the size of the workforce being 75 at the time. Digital was one of the first firms in the industry to accept venture capital. Kildall once said: "It feels good to know that my program has inspired other authors to write best sellers, but with all the other overnight successes, people tend to forget that you're bound to have some flops. I wrote 18 programs on speculation before I came up with my first hit."

2.4 Visicorp

Visicorp was founded in 1978 as Personal Software Inc. in Boston USA by Daniel Fylstra, a BSc Electrical Engineering & Computer science graduate from MIT, who was completing his MBA at Harvard, and Peter Jennings, a BSc and MBA graduate of McMaster university of Ontario. The company was established to market the 'MicroChess', program, the first microcomputer chess-playing program, which was written by Jennings (15).

The program was an immediate success and over 50,000 copies were sold, making it one of the first microcomputer software products to gross over a million dollars in sales.

At the time of the launch of 'MicroChess', Dan Brinklin, a classmate of Fylstra, had developed the idea of a "visible calculator", a program which would perform calculations similar to a financial spreadsheet. Since Personal Software had a good reputation as a software marketing company, Briklin approached Fylstra with his idea. Fylstra was so impressed by the concept, that he loaned Brinklin some money as well as an 'Apple II' computer on which to develop the program. Brinklin joined forces with Bob Frankston, a friend from MIT, and formed a company called Software Arts which was to develop the now famous 'VisiCalc'. The program took almost a year to develop from the prototype, and the finished product was launched in October 1979.

Brinklin relocated Personal Software to Sunnyvale, California in mid 1979 and was granted sole marketing rights for the product in terms of the aggreement with Software Arts.

'VisiCalc' was immediately accepted as the top software product for the 'Apple II' and sales incresed rapidly. Personal Software was founded with an initial cash investment of $500 and had been financing its activities through the Bank of America. In May 1980, the company raised two further investments of $250,000 each from Venrock Associates, the venture capital arm of the Rockefeller family and Arthur Rock, who had helped found Intel Corporation. The company moved into new premises of 13,000 sq ft in Sunnyvale and in September 1980, Personal Software hired its first outside professional manager, Terry Opdendyk, from Intel, as President and Chief Operating Officer. In March 1981, a second round ov financing was announced with the private placement of a further $2.1 million of the company's stock with six venture capital groups, Venrock Associates, Lamoreaux, Glynn & Associates, Fayez Sarofim & Co., Asset Management, Newcastle Company and David Karetsky.

During its first three years, the company served both the personal computer hobbyist as well as the business markets. With the tremendous growth of personal computer usage in business, the company decided to discontinue its home entertainment, education and game product lines in mid-1981. The range was rapidly expanded with the announcement of the 'VisiSeries' products 'Visifile', 'Visidex', 'Visiplot', 'Visitrend' and 'Visiterm'.

In November 1981, the company moved to premises covering 78,000 sq ft in San Jose and in January 1982, its first international marketing office was opened in Paris. At this stage, the number of employees was 95.

In February 1982, the name of Personal Software was changed to VisiCorp to associate the company more closely with the highly successful 'VisiCalc' program and the 'VisiSeries' family of productivity software. In mid-1982, 'VisiLink' was released. This was a joint development with Data Resourses, a subsiduary of McGraw Hill Inc. Further enhancements were made to the original 'VisiCalc' program to produce versions in German, Italian, Swedish and French. 'VisiSchedule' was released.

In December 1982, the 'VisiOn Open Application System' was announced, incorporating an interactive software environment that allowed a personal computer user to work with a number of applications simultaneously with windows and a mouse.

1982 Revenues were $35m, more than any other microcomputer software company (16).

In early 1983, VisiCorp released 'VisiWord' and 'VisiSpell' for the IBM Personal Computer. In March 1983, three further operations were announced: 'VisiPress', a book publishing operation, 'VisiTutor', a series of computerised training courses and 'VisiTraining', a series of professional seminars (17).

In April 1983, VisiCorp acquired Communications Solutions Inc. and in May 1983, a joint development project was initiated with Informatics General Corp to produce 'VisiAnswer' and 'Answer/DB'. In the same month, a three way announcement was made between VisiCorp, Xerox Corp. and 3Com Corp. to allow 'Visi On'X users to utilize the 'Xerox Ethernet LAN'. By mid-1983, Texas Instruments, Wang Laboratories, Xerox and Honeywell had all indicated their intention to support the 'Visi On' system and in July, Applied Data Research announced its intention to develop its mainframe software to support the 'Visi On' system.

In the third quarter of 1983, 'VisiCalc IV' was released in conjunction with 'Multisoft', adding graphics to 'VisiCalc' on the IBM PC.

Revenues for 1983 were $43m, 20% higher than the previous year, a relatively low increase in an industry where 100% growth rates are more the norm.(18)

In January 1984, Dan Fylstra was asked whether he was nervous about betting his company on 'Visi On'. His reply was "Well I'd like to quote Steve Jobs about this. He was asked a similar question at a forum and said that "if 'Lisa' doesn't make it, then we'll (Apple) just be another half billion dollar company " (19).

Soon after this, VisiCorp experienced financial problems. In May 1984 the company laid off 45 people, and drastically cut prices on all its programs. Prior to these actions, its workforce had already been reduced from a peak of 270 to 55 and several expansion projects had been abandoned. An announcement was made that VisiCorp was to reposition itself to sell to home and business markets, a far cry from all its previous ambitious plans (20).

In November 1984, VisiCorp sold its communications subsidiary, 'Commun cations Solutions' as well as 'Visi On', to Control Data. VisiCorp merged with Paladin, a California software house, retaining the marketing rights for the "Visi" products. Paladin's president, Roy Folk, was appointed head of the new operation, and VisiCorp chairman and chief executive, Daniel Fylstra, resig.ed from his post.

At the time of writing, VisiCorp products have been slashed to less than a quarter of their original prices in an effort to regain popularity.

The name "VisiCorp" has disappeared as a dominant force in the industry in less than one year!

2.5 Software Arts

Software Arts was founded in 1979 by Dan Brinklin, then 27 and Robert Frankston, then 29. In the spring of 1978, while Brinklin was completing his MBA at Harvard, he had a better idea for a way to "crunch" numbers. "I visualized an electronic blackboard and electronic chalk in a classroom," he said (21).

In the summer of 1979, Brinklin formalized the programming problem on paper and produced the first working version which was limited to 5 columns and 20 rows and was difficult to use without some knowledge of programming. Realizing that he needed someone who had a better understanding of machine code, he teamed up with Bob Frankston, whom he had met at MIT in 1970. Together with

Brinklin and Steve Law, their first employee, Frankston managed to design the complete VisiCalc program into less than 20,000 bytes of memory.

At this stage, Brinklin was introduced to Daniel Fylstra, on whose suggestion they began coding the program for the Apple computer. Fylstra expressed an interest in selling the as yet unnamed 'VisiCalc' together with his other software packages through Personal Software.

Brinklin and Frankston formed Software Arts in January 1979 and spent the next nine months coding the program. Alternating between university lectures and program development, the two pioneers completed the program, working mainly at night in Frankston's apartment on a time sharing terminal since rates were cheaper.

In April 1979, Fylstra decided on the name 'VisiCalc', which stands for VISIble CALCulator, and for which Personal Software held the rights. In May, Brinklin and Frankston showed a prototype of 'VisiCalc' to computer dealers but failed to impress them. In June, after Brinklin had graduated, the two gave a presentation at the National Computer Conference in New York but did not attract more than a dozen people. In July, they managed to scrape together $20,000 and raise a loan of $65,000 and bought a 'Prime 550' minicomputer for development purposes. That month, they received a rave review by Benjamin Rosen, a respected computer industry analyst, in his newsletter when he described the package as the "software tail that may wag the personal computer dog."

'VisiCalc' was finally introduced in October 1979 and revenues in 1980 were $1m, followed by over $3m in 1982. In terms of the aggreement, Personal Software was required to pay Software arts a royalty of 36 to 50% of its revenues from 'VisiCalc' (depending on how the program was sold) (22). The number of employees at Software Arts grew from 4 to 35 in twelve months and in 1982 the staff consisted of 50 people. The reason for the dramatic growth was that 'VisiCalc' was "the first program that proved that personal computers could be useful for business people (23)."

In 1981, Brinklin was awarded the 'Grace Murray Hopper Award' by the 'Association for Computer Machinery for Significant Accomplishment in Computers by People under 30'.

More recently, Software Arts developed a software package called 'TKI Solver' package for engineering and mathematical applications.

2.6 Lotus Development Corporation

Mitch Kapor founded Micro Finance Systems in his apartment in Watertown, Mass after he bought an Apple computer on impulse. He decided to drop his Ph.D in psychology and go into business programming computers instead. In 1981, he sold his 'VisiPlot/VisiTrend' program to Personal Software for $1m and used the money to start a software development company, Lotus Development Corporation" (24).

The advent of the 16-bit micro with its larger memory size and greater disk capacities heralded the development of a new generation of software, the integrated package. The integrated package is a multifunctional package, combining multiple applications in one program.(AR/CW) 'Lotus 123' was the forerunner of this new development.

'Lotus 123' achieved sales of $53m in its first year (1982/83) (25). It has been said that "'Lotus' did for the IBM PC what 'VisiCalc' did for the Apple" (26). More recently, Lotus introduced 'Symphony', a more powerful version of its predecessor, "123".

2.7 Paul Lutus

Paul Lutus wrote the popular 'Apple Writer' wordprocessing program in a plywood cabin on top of a mountain in Oregon, 23 miles from the nearest town. He had to string up a 1300 ft extention cord to get electricity to run his computer (27).

In 1983, Lutus earned $2,8m from royalties from his program which is among the top selling of all wordprocessing programs on microcomputers. Today, Lutus lives in a "spacious $240,000 home on 35 acres on his own mountain in southwestern Oregon" (28).

2.8 Software Publishing Corp

Fred Gibbons founded Software Publishing Corp in Mountainview, in 1980 in California with $250,000 in venture capital. Gibbons' programs, the 'PFS suite', were extremely simple to use, yet sufficiently powerful to perform business functions. By 1984, Software Publishing was shipping between 35,000 and 45,000 units per month (29). Software Publishing recently entered into an aggreement with IBM to develop the 'Assistant' series, which is basically the 'PFS' range with the IBM logo.

3. TRENDS WITHIN THE INDUSTRY

3.1 Introduction

Although some software companies are six or seven years old now, it may be argued that the industry really began to bloom when Visicalc was released over five years ago, opening the door to a new consumer market blending marketing strategies with technical excellence (30). Initially no one knew exactly how to run a software company. Before personal computers, a software hit sold a maximum of 1000 units and used to sell for $50-60,000. Now with programs like 'Visicalc', 'Wordstar', 'dBase II' and 'Applewriter', companies can sell hundreds of thousands of units for prices ranging from $50 to $700.(31)

3.2 Early Development

The early application programs were developed by personal computer hobbyists who were possibly more interested in showing off their work than making money from their programs. Copies of programs were freely traded at computer club meetings (32). As recently as 1980, software was very much an amateur cottage industry, with programs packed in plastic bags with badly written, oft3n photocopied instructions, and sold through the mail.

3.3 The Industry Matures: 1980 - 1983

As is evident from the above overview of the early software pioneers, the 1970s saw the development of professional sofware companies, established by young entrepreneurs, usually in their mid-twenties. These companies were initially undercapitalised and all raised finance through venture capital. All seem to have suffered from a common problem - lack of business skills due to young, inexperienced management. Early companies like Visicorp, Digital Research Micropro and Microsoft made plenty of mistakes in their strategies They had no real idea of how to make realistic projections of sales or market direction and did not have any historical references by which to make sound decisions (33).

The early 1980's, saw the industry taking a definite shape as companies started changing their strategies to fit the market in such key areas as product positioning, professional management, compatibility, marketing and distribution. It may be postulated that an important area in strategy planning in any software publishing company, is the product life-cycle around which the company plans its product introductions, marketing strategies and research and development. Furthermore, it may be argued that until recently, the software business had simply not been around long enough to know exactly how long a product's life cycle would be. 'Visicalc' was the first successful product to have gone through a complete life cycle, from conception in 1978 to introduction in 1979 to peak success in 1982 to decline in 1983 and to possible death in 1985 (34).

This period also saw the growth of the independent programmer, operating much like a novelist, selling software to publishers in return for a percentage of the sales. Over fifty of these freelance programmers have earned more than $500,000 each (35).

3.4 Consolidation: 1984 - 1985

It appears that 1984 has spelt the death of the independent programmer as the overnight successful software house. As early as 1981, Mitch Kapor, founder of Micro Finance Systems, said: "A number of pioneers in this industry have gotten rich in the past few years by being merely competent and very lucky. The costs of team development ranges from $250,000 to $500,000, so fewer independent authors will be able to compete (36). It is also evident that the ever - expanding number of software titles in existence has made it far more difficult for the independent programmer to make an instant success with a program. Where royalties were as high as 37% of the wholesale price, these have dropped to below 10%. (37).

Most of the latest software is far too complex to be written by one individual. Most software houses are using development teams ranging in size from 8 to 24 people.(38)

This period has seen a degree of consolidation in the industry. The folowing effects have become apparent:

* some small software companies have already been reorganised under bankruptcy laws.
* other small software companies are cutting back on operations and laying off employees.
* some larger well known software companies are experiencing cashflow problems and are being acquired by larger companies.
* many companies, both large and small, are up for sale.
* over the past year, about two dozen micro software companies were acquired by a wide range of companies from other industries, including mainframe software, book publishing and even entertainment.(39)
* It is apparent that respected venture capitalists and institutional investors are becoming increasingly sceptical about investing in software companies.(40)

Some companies have, however remained stable. These include Lotus Development Corporation, Software Publishing and Microsoft.(41)

This period has seen the entry of giant corporations from other fields into the software industry. It may be argued that the attraction of the software industry lies in its potentially huge profit margins. "Programs that cost only $5 or $10 to make can retail for up to $700."(42)

CBS, Walt Disney Productions, 20th Century Film Corporation and Dow Jones & Co. all entered the software business during 1983. CBS and 20th Century are producing games software, Disney educational programs and Dow packages for stock market analysis.

The 'Readers Digest', which sells approximatelv 30m copies of its magazine worldwide per month, launched its first software products at the Consumer Electronics Show in Las Vegas in March 1984. The publisher is banking on its "easy to read reputation spilling over into its software offerings which are also easy to use." (43)

Every major publishing firm in the USA is currently planning to release software programs or has already done so (44). Successful entrants in the market include Wiley, McMillan and Hayden. American Express has started to mail software catalogues to its base of card holders. According to Bill Ablondi of Future Computing, "it makes all the sense in the world for publishers to come out with software. The content and distribution could be very similar to books."(45)

IBM, almost 10 times the size of its closest competitors in the areas of hardware and non - micro software, could be in a similarly dominant position in the micro software domain in 5 years time with the release of an integrated portfolio of low priced commonly used applications in mid 1984 (46). Bearing the IBM logo, the portfolio includes a wordprocessor and spelling checker, database, business graphics, and spreadsheet.

With the entry of the larger companies into the business, the whole industry has become increasingly marketing-oriented. In fact it is becoming similar to the "record business" (47), with companies aiming for that "monster hit that sells thousands of copies (rather) than design specialized programs that appeal only to tiny segments of the market" (48).

It would seem that the entrance of the large company into the business will make it hard for the smaller companies to compete, and will probably spell the death of the individual programmer/entrepreneur of the seventies who produced, packaged and marketed his own product from a backroom. In the words of Daniel Fylstra, chairman of Visicorp, "Before, anyone with a reasonable product could make a go of it. Now you're seeing larger and larger sums directed towards marketing. Brand names are becoming more and more important."(49)

4. DISTRIBUTION METHODS

4.1 Introduction

With the development of the personal computer in early 1975, the industry witnessed the birth of the "computer shop". The first computer shop was opened on 1st July 1975 by Dick Hizer in San Harmonica, California (50). At time of writing, approximately 4000 computer shops are in existence in the USA (51). The 1980's saw the development of the "software only" store, which according to Portia Isaacson, president of Future Computing, should capture 25% of the projected $8bn software market in 1987 (52).

While the retail store remains the primary method of distribution of software to the consumer, it is the author's contention that alternative methods of distribution will be adopted during the latter part of the decade. Possible alternatives to the above method of distribution are as follows:

4.2 Electronic Distribution

Tymeshare, a computer telecomunications firm in California, is experimenting with the electronic delivery of programs to customers. After the purchaser has selected his program in a store, the salesman orders it from a central computing facility.

The software is copied onto disc in the store, debiting the store's account automatically (53).It may be speculated that this type of distribution will occurr in the home itself in the near future. This process is also being developed by Softyme Express in conjunction with Micro D, a computer distributor to place reproduction machines in stores. Two additional firms testing or marketing similar systems are Romox of Campbell, California and Xante, of Tulsa (54).

An outline of research in regard to telesoftware research performed in the United Kingdon is presented in a handbook produced by the 'National Computing Centre' or NCC in that country.The NCCUK has performed a considerable amount of research in regard to "telesoftware" on a national basis as a public service. This involves "storing computer programs in a centralized base and down-loading them via a telecommunications link for execution on a remote computer - usually a micro - on an as required basis."(55) The Telesoftware concept is an extension of an information Database such as 'Prestel', consisting of a Sponsor to provide initial funding, Information Providers to maintain the information on the database, the Facilities Manager controlling the storage and technical facilities, and the Carrier which connects the database to its users. The users,in turn, pay for the use of the software.

In 1979, the 'Council for Educational Technology' (CET) in the UK developed a process for distributing programs to schools for use on the 'RML 380Z Schools Computer'. The software was stored on 'PRESTEL' and distributed via the Public Switched Telephone Network. The standards developed by CET have become the de facto 'PRESTEL' Telesoftware Program Database standards. 'PRESTEL' has subsequently been building up its own library of software, augmented by compilations from other publishers such as Practical Computing.

According to the NCC, the Sponsor for a Program Database along the lines of 'PRESTEL' could be from the publishing industry or a major software house. The Information Providers would consist of software houses and academic institutions. The Facilities Managers would be organizations set up by the sponsors for the purpose of bulk storage of programs as well as communications. The Carrier would be British Telecom initially, with interactive cable and radio as alternatives for the future.

At the time of publication of the NCC booklet, no real interest in Telesoftware seems to have been evoked in the industry.

4.3 Vending Machines

Nolan Bushnell, multimillionaire founder of 'Atari', started a company in 1983, Cumma Technology Corporation, to develop a software vending machine. The product, called 'Metawriter', is operated by coin and holds a large range of programs on a

'Winchester' hard disk. The prototype, introduced in early 1984, held 70 programs and dispenced software out of four slots for 'Atari', 'Commodore VIC20', 'Commodore 64' and 'Coleco Adam'. The customer inserted a $5 re-usable program cartridge in the slot, deposited between $1 and $20 for the software, and received the program together with documentation produced by a small printer. Plans for floppy disk - based machines were also on the drawing board. The intention was to place machines free of charge in computer stores and mass merchandisers, passing on a royalty of 20 - 30% for each program sold. "Cumma" embarked on a campaign to attract independent authors to submit software, offering an average royalty of 15%.(56)

4.4 Vertical Packages

An upscale vertical package, where the frequency of sales is very low, and where stocking the product may be prohibitively expensive, is often difficult for the software author to sell (57). The software which falls into this category includes plumbing, farming and medical applications. To solve this problem Intertec Data Systems Inc in California introduced 'Dial A Demo' in 1984, a free service where dealers could call up 24 hours a day to show customers a demonstration of an 'Intertec' package. If the customer was happy with the on-line demonstration, the package could be ordered immediately and the dealer would subsequently receive a commission for the transaction (58).

4.5 Broadcasting

During 1981, The British Broadcasting Company (BBC) in the United Kingdom started an experiment to transmit software for use on an 'Acorn' (the BBC computer) equipped with a special decoder. (59) In January 1984, the BBC began "broadcasting" software on the airwaves to listeners on the program 'Chip Shop'. The software was broadcasted in 'Basicode', which originally had been developed for a Dutch radio program involved in a similar transmission. A conversion program, also called 'Basicode', was available for different makes of computer, at a cost of R7, to translate the software recorded in audio form from the BBC into a usable form for that particular computer. Initially, program material was confined to games, but plans to expand the range will depend on the success of the initial project.(60)

4.6 Public Domain Software

The increasing utilization of the microcomputer by computer enthusiasts has led to the formation of numerous user groups throughout the world. Collectively, these groups may be referred to as the "Public Domain". Each club has a percentage of technically minded members who develop their own software for free distribution among their friends. The concept of a public domain network involves storing these programs on a database and making them accessible to members through telephone links. The majority of these groups exist in the USA and various publications are available with telephone numbers and connection details of members. Any individual owning a personal computer may dial into the database, which may be simply another micro at the house of a member of the group, and can then retrieve programs free of charge. Some of the larger groups are more structured and charge a membership fee. Different groups exist for all makes of personal computers.

4.7 Freeware

A new concept of software distribution has evolved with the development of "freeware" or user supported software,(61) serviced by the numerous computer - user groups throughout the USA. These associations compile catalogues of programs written by amateur software authors, usually group members. Users are permitted to make copies of these programs and if they decide a particular package is of value to them, they are asked to send a suitable donation to the author. Typical donations range from $3 to $75 per program. This concept offers authors the opportunity to break into the software business without making a big initial investment. "Freeware" is distributed in the same way as public domain software - through user groups, computer clubs and remote bulletin board systems. The author of a program called 'PC File' has earned over $100,000 through user submitted contributions, and there appears to be an equal division between those those who contribute and those who do not.(62)

5. SOFTWARE EVALUATION TECHNIQUE

5.1 Introduction

The ever increasing explosion of software available to the consumer makes the task of selecting the best package for a particular application extremely difficult. The author recalls in the early days of the industry in the late 1970's, when it was possible to know in detail, most of the software on personal computers in the country. Nowadays the proliferation of range makes this impossible, but then the quality of both the software and associated documentation has improved to such an extent that the purchaser - now very often a second time user - does not require the same degree of assistance as in the past. How then, does one go about selecting and supporting the "right" wordprocessor, spreadsheet, or any other package for that matter? In the author's opinion, no structured evaluation technique exists in published form. Computer journals such as "Computerworld" and "Byte" regularly publish evaluations of ranges of software packages within specific applications, but the basis of analysis is limited to checklists of features. The author has developed the following structured technique over the past four years specifically for the retail trade, although the approach could easily be modified for any other facet of the distribution cycle.

5.2 Methodology

The evaluation process may be divided into five distinct steps (see fig.1)

- STEP 1 - generic classification
- STEP 2 - application selection
- STEP 3 - package evaluation
- STEP 4- product positioning
- STEP 5 - support systems definition

The technique involves utilizing the concepts outlined in the five steps above as well as the appropriate tables and checklists to evaluate software in the context of the organization's specific business policies.

5.3 STEP 1 - Generic Classification

5.3.1 Introduction

The nature and range of software packages is extremely diverse; packages exist from simple games costing less than twenty rand up to complex integrated business systems costing thousands of rands. As a first step within the structured selection process, a classification system is suggested which divides "software packages" into six major generic categories as follows:

* Games, hobby, some utilities
* Simple package
* Complex package
* Complex system
* Application generator
* Custom software

This generic division is summarised in Fig.2. Factors to be considered in this delineation of software include system design, documentation, education requirements, installation requirements, ongoing support requirements, user level and pricing.

Each generic category will now be examined with regard to the above factors.

5.3.2 Games, hobby, some utilities

This category comprises the vast array of low-end software made up of games and utilities for the hobbyist and enthusiast. Many of these games would have been written previously on the Apple and subsequently transferred onto other microcomputers. These packages require the minimum of documentation over and above start up instructions and require no education or installation support. Similar to arcade games, TV games and records, these packages often display a relatively short life cycle. Prices of these packages are usually below R100.

5.3.3 Simple package

The simple package is identified by its characteristic in performing a specific task effectively within the limitations of its design, taking into account associated documentation, as well as the fact that the user could possibly be computer illiterate. Examples of simple packages are "Visicalc", "Applewriter", and "Omnis". These packages perform the tasks of financial modelling, wordprocessing and filing respectively. The simple package may not be configured for any specific user environment or hardware combination alternatives; it simply "works" as described in its user guide. The simple package may be viewed as a consumer item, with sales or potential unit sales running into tens, or sometimes hundreds of thousands. Its design and documentation is at a mature level; the software is proven and free of bugs and its user manual is usually of fine quality.

Users of the simple package display a degree of competence within the scope of their jobs involving the computer, and make use of the computer as an automated means of achieving something with which they are professionally fully conversant. Such people are highly motivated to understand the system and are usually prepared to spend much of their personal time (initially) learning and experimenting with the package. Examples are the secretary using a wordprocessor and the financial manager manipulating a spreadsheet. Assuming adequate documentation is provided, training and educational requirements are minimal. Support is required mainly on the telephone; problems encountered are invariably as a result of "finger trouble". Prices of the simple package are usually between R50 - R1000.

5.3.4 Complex paokags

The complex package usually performs a range of functions and by definition is more complex in nature than the simple package. Examples oi complex packages are accounting systems and integrated software packages such as Lotus 1,2,3. Complex packages often need to be set up by the dealer or configured for user-specific application and hardware conditions. While the documentation of the complex package is usually highly developed, the user manual alone will not be a sufficient substitute for training, and additional education will need to be provided. The complex package is often handled by an operator such as a clerk or secretary with a limited understanding of the prcblem to which the system is being applied. This person may display an initial fear of the computer and tend to operate the system only by following every step in the manual, consequently requiring a considerable amount of initial training and subsequent support. Ongoing support is required both on the telephone as well on a personal classroom basis and will include general package assistance as well as operator retraining to cater for staff turnover at the user site. Pricing is usually in the range of R500 - R3000 and may include the cost of installation and initial training.

5.3.5 Complex system

The complex system performs the function of a "business system", as opposed to a package which fulfills a "single function" within the business. Examples of complex systems are manufacturing, medical and agricultural systems. The requirements of a business system are usually too unique from one business to another to allow for the simple configuring of a standard package to meet all of its needs, and in many cases cannot be satisfied by imported packages since local business practices differ from country to country. In the case of a business system, the dealer requires specific expertise in the application under consideration, in addition to technical computer knowledge. Complex systems are usually written locally by software houses and are sometimes termed "vertical market packages", since they cater for specific markets. Due to their system complexity and limited unit sales, many of these complex systems have not yet fully matured and exhibit invariably lower quality documentation and minor operational problems which have to be resolved by the relevant software house. A complex system may become a complex package as the market matures and the quality of the package improves. Complex systems need to be installed for the user, and considering the wide range of people operating the system, a substantial amount of initial training will have to be provided. On-site, classroom as well as telephonic support must be provided as an ongoing service. Pricing of complex systems is usually R1000 - R15000 and includes installation and initial training.

5.3.6 Application generator

If the requirements of a business cannot be met by an existing package or complex system ("vertical market package"), a unique solution may be written by a system builder. The system builder makes use of a fourth generation language technique such as dBase II, Condor or Generex to develop the system. The system builder often uses previously developed system components or macros to speed up the process The system builder must develop an operator manual with the user. The requirements for education, installation and ongoing support apply as in the case of the complex system. Systems developed in this mannor often become "vertical market packages". Pricing of application generated solutions vary according to the complexity of the system and include the cost of the appropriate fourth generation language.

5.3.7 Custom software

Custom software is an alternative to application generated stems and will be resorted to if the application is of such a nature that a fourth generation language will not suffice or if there is a specific need to program in a certain language for interfacing to other systems. In some cases, users may possess … programming capability with which they would like to write the system. Support and pricing details apply as in the case of application generators.

5.3.8 Trends within the matrix

The software package market is in its infancy but is evolving rapidly. As it matures, certain trends will become apparent with reference to the generic classification table. The trend will be for software to move up the tab'e, with complex systems evolving into simple packages, as consumer awareness and computer literacy improves. Systems which are considered complex by today's standards will be transformed into standard packages in the future, being replaced by even more sophisticated systems which possibly are currently not yet implemented on microcomputers.

5.3.9 Conclusion

It is suggested that the dealer's market philosophy be analysed in accordance with the principles detailed above. A decision may then be made on which areas to concentrate within the parameters of available personnel resources.

5.4 STEP 2 - Application Classification

5.4.1 Introduction

The range of applications and systems covered by the software package market is all-embracing; theoretically, any business system or application which is performed manually can be converted into a software package and the drive towards automation continuously fuels the development of new software packages.

For the purpose of this analysis, the range of applications has been divided into five main categories:

* Entertainment
* Business and commercial
* Scientific
* Technical
* General

Each category may be further divided into specific application areas as follows:

5.4.2 Entertainment

This group consists mainly of low priced, non-support items which are highly seasonal in nature.

- childrens games
- business games
- games
- educational games
- hobbies

5.4.3 Business and commercial

The majority of software packages fall into this grouping. Specific application skills are a prerequisite for selling in each application area.

- financial modelling
- business graphics
- database
- filing systems
- stock market
- wordprocessing
- mailing
- accounting
- vertical market applications

Vertical market applications are industry specific applications, eg production engineering, medical, farming etc., requiring an in-depth knowledge of the relevant topic on the part of the dealer.

5.4.4 Scientific

The scientific grouping consists of products with relatively long life cycles requiring extremely specialized application expertise.

- mathematics
- statistics
- engineering
- C A D

5.4.5 Technical

Operating systems, utilities and languages make up this group, which also exhibits relatively long life cycles due to hardware dependance.

- utilities
- operating systems
- compilers
- languages

5.4.6 General

This group contains all products which do not fall into any of the previous groups.

- communications
- training aids
- training
- education
- graphics

5.4.7 Conclusion

It is suggested that the dealer's application skills be identified in terms of manpower resources in accordance with the above classification. It is advised that where related application skills do not exist, concentration be channelled into more familiar areas.

5.5 STEP 3 - Package Evaluation

5.5.1 Introduction

The package evaluation checklist is intended as a qualifying test to provide an indication to the evaluator as to whether it will be worth spending further time reviewing the functional content of the program. The content of the suggested checklist is general and as such may be applied to any software package, irrespective of application. The checklist is in the form of a series of questions, and it is for the evaluator to interpret the results with appropriate weightings.

Since the evaluation process can be extremely time consuming, it would be preferable to structure the checklist in such a way so as to provide the evaluator with a means of early detection of poor package performance. An example of an indicator could be the current existence of a superior, cheaper product on the market. The evaluator could decide at this point whether or not to reject the package, thus obviating the need to spend considerable effort in completing the review cycle. To this end, the checklist has been further divided into an initial screening, general screening and functional screening. The initial screening has been designed as a quick method of package review without the need for running the software. If the program falls short at this stage of the test, the evaluator should consider rejecting the package to conserve further effort. The general screening covers the detailed practical operation of the package while the functional screening provides a measure of the excellence of the application itself.

5.5.2 Initial screening

Marketing details

* Is there a market for the product?
* Is there a need for the product?
* Are there competitive products?
  - how do they compare in terms of price and function?
* Are there better products available?
* What makes the product special?

Manufacturer's details

* What is the manufacturer's history?
* How long have they been in business?
* What other software products do they supply?
* How is the product performing in the USA and UK?
* What is the market share in the USA and UK
* What is the market acceptance of the product in S Africa?

Distribution details

- What is the local supplier's history?
- Can the local supplier support the package?
- Can the local supplier demonstrate the package?
- What is the financial strength of the local supplier?
- Will the local supplier keep stocks?

Technical details

- What is the upwards growth potential of the package?
- Does the package support locally available peripherals?
- Does the package support hard and floppy disks?

5.5.3 General screening

User interface details

- Is the package easy to install?
- Is the package user friendly?
- Is there an interactive help facility?
  - how effective is it?
- Is the software menu driven?
- Is the program easy to use?
- Does the software make use of function keys?

Documentation and manual

- How good is the quality and appearance of the printing?
- What is the size and appearance of the binder?

* Is the text easy to understand?

* Is the manual accurate with reference to the software?

* Is there a table of contents?

* Is there an index?

* Is there a glossary?

* Is there a quick reference card?

* Is there a tutorial lesson?

* Are there built-in examples?

* Is there an error message list?

Packaging and presentation

* Is the packaging sturdy?

* Does the packaging have the ability to still look new after being opened?

* Can the package stand squarely on a shelf?

* Is the packaging attractive?

* Does the package contain information about the product?

Program performance

* How fast is program execution?

* How is the overall performance?

* Are there input validation checks?

* Does the software perform bug-free?

* How easy is it to recover from errors?

* How useful are the error messages?

* Is the text easy to understand?

* Is the manual accurate with reference to the software?

* Is there a table of contents?

* Is there an index?

* Is there a glossary?

* Is there a quick reference card?

* Is there a tutorial lesson?

* Are there built-in examples?

* Is there an error message list?

Packaging and presentation

* Is the packaging sturdy?

* Does the packaging have the ability to still look new after being opened?

* Can the package stand squarely on a shelf?

* Is the packaging attractive?

* Does the package contain information about the product?

Program performance

* How fast is program execution?

* How is the overall performance?

* Are there input validation checks?

* Does the software perform bug-free?

* How easy is it to recover from errors?

* How useful are the error messages?

Program effectiveness

- Does the program accomplish its stated objectives?
- Does it perform the task effectively?

Marketing details

- Will there be local advertising?
- Is the pricing satisfactory?
- Is the margin satisfactory?
- What is the multiple disk policy?
- Is free demonstration material available?
- Is a free demonstration package available?

General

- Can the software integrate with other products?
- Is the software available under other operating systems?
- Is the software copy protected?
- Is end-user training available?
- Is there a hotline for end-users?
- Will updates be provided? At what cost?

5.5. Functional screening

The functional screening is a listing of functions pertaining to a particular application, grouped into functional categories. The specification is meant to serve as a guide for evaluating the functional capabilities of a package. It must be borne in mind that the mere fact that one package consists of a larger number of functions than another does not necessarily mean that is is superior. The application specification is not constant with time; it varies dynamically as technology increases the capabilities of associated application packages. The listing should therefore be initially tailored by the evaluator to suit his particular application needs and thereafter maintained on an ongoing basis as new developments take place.

It is up to the evaluator to decide which functions are essential, useful or superfluous, since there is no absolute scale for measurement.

he following listings have been prepared as examples (see Appendix A):

- Wordprocessing
- Database
- Spreadsheet (financial modelling)

5.5.5 Conclusion

It is suggested that the software package under consideration be subjected successively through each of the three screening processes outlined above; cognisance must be taken of the fact that the relative excellence or otherwise of a package is a subjective measure, checklists being used merely as a reference.

5.6 STEP 4 - Package positioning

5.6.1 Introduction

Unlike the case of a conventional retail line, taking on a software product is more than simply placing a price tag on the item and allocating shelf space. Software is intangible; its features and benefits can not merely be understood from a cursory glance at the floppy disk or by skimming through the associated manual. Taking on a software package in the true sense requires a committment on the part of the retailer in terms of the time involved in getting to know the system thoroughly. It is seldom possible to become 100% familiar with a software package of any complexity through any amount of presales review; most non-standard queries and exceptions arise at a later stage once the package has been sold and is subjected to practical usage. A complex package such as "Lotus 123" or "dBASE II" can take many months of concentrated effort to master.

It is suggested that the average computer dealer does not have the time to become familiar with every package in his range. This may often result in unhappy customers and under-utilized software systems. This state of affairs is largely due to the fact that software packages are currently "high fashion" items with a very short product life cycle, often less than six months. The general attitude is often "why should I spend time getting to know a package if I may no longer stock it in a few months time?"

A possible solution to this problem is to position the product into four broad categories as follows:

* Fully supported, intimately known packages
* Supported, lesser known packages
* Unsupported packages
* Games and certain utilities

5.6.2 Fully supported, intimately known packages

It is suggested that the retailer choose certain key packages from each product category (ref. 5.4) in accordance with the application skills of his staff and his company philosophy. These packages should be totally dissected and understood. Very often the expertise in a certain package may be carried by a combination of individual talents and not be centred around a single person. The selection of packages in this category will possibly determine the very existence of the retailer and the development or decline of his business. The 80-20 "rule of thumb" applies; the majority of systems sold will ideally be based on these packages, and in some cases these packages may even constitute the total product range of the retailer. The retailer should become a specialist in these applications.

In accordance with the above principles, a prime consideration for taking on a package, is application expertise within the business. The number of packages and product categories chosen will therefore depend on the application skills and number of people in the business.

5.6.3 Supported, lesser known packages

This category consists of a wide range of packages which have a place in the market but which are not "best sellers". It is suggested that packages in this category should only be stocked if a degree of related application expertise exists in the company. Familiarisation should involve only a brief examination of the manual and a quick review of the software; exceptions and queries will be handled as they arise. As expertise develops, certain of these packages may even be transferred into the "fully supported" category.

Very often, packages in this category will be held in stock in the form of demonstration models at no cost to the retailer, orders being placed only after the sale has taken place.

5.6.4 Unsupported packages

This category comprises packages that are either too "application-specific" or too technical to be understood or supported by the retailer, but for which a demand nevertheless exists. Familiarisation should involve merely understanding the objectives of the package.

Where possible, demonstration copies should only be kept in stock.

5.6.5 Games and certain utilities

These comprise low priced packages that require no support whatsoever. Games in particular are highly seasonal in nature. Sales of games very often depend on the current "fad" and also on pakaging and presentation. Utilities are less seasonal and due to their technical nature will usually only be purchased by the hobbyist.

5.6.6 Conclusion

It is suggested that each package in the retailer's range be allocated to one of the above four categories and the familiarisation process be adhered to, in accordance with the above guidelines.

5.7 STEP 5 - Support Systems

5.7.1 Introduction

It is suggested that many computer dealers rely for their customer support services on one or more individuals referred to as "support people". Support is seldom charged for as dealers may be embarassed to demand payment for services which are often of inferior quality. This is due to the fact that support is usually provided by individuals who may be computer literate but who may have little experience in business or in the specific application at hand.

In light of the above, the support department often never develops; it is usually run purely as a cost centre as a necessary part of the business.

5.7.2 The support department

In the author's opinion support services should be charged for in the same manner as a Medical Practitioner charges for his time. The support department in the business should be run as a separate profit centre and should be entirely self funding. The support department will thus grow and develop with the business itself and even add to its profitability.

To make allowance for staff turnover in this vital area, it is necessary to assign individuals on a primary and secondary basis to each application area with which the business is involved. This method results in the development of "application specialists" rather than "support staff". In a larger organization, the degree of specialization will be easier to achieve, while in a smaller business, various applications may be shared by suitable individuals. (fig 3 illustrates this point.)

5.7.3 Conclusion

It is suggested that once the retailer has decided on the applications with which to become involved, resposibility for support be assigned to certain individuals on a primary and secondary basis to these areas.

with the business itself and even add to its profitability.

To make allowance for staff turnover in this vital area, it is necessary to assign individuals on a primary and secondary basis to each application area with which the business is involved. This method results in the development of "application specialists" rather than "support staff". In a larger organization, the degree of specialization will be easier to achieve, while in a smaller business, various applications may be shared by suitable individuals. (fig 3 illustrates this point.)

5.7.3 Conclusion

It is suggested that once the retailer has decided on the applications with which to become involved, resposibility for support be assigned to certain individuals on a primary and secondary basis to these areas.

APPENDIX A

FUNCTIONAL SPECIFICATION

WORDPROCESSING

GENERAL

Years on market

Menu driven

Displays text as it will
appear in print

Prints one file while
editing another (spooling)

Edits programs and text

Handles f'les larger than memory

FILE CONTROL

Continuous back-up

Automatic back-up on file save

Save file and continue editing

Insert second file

Insert portion of second file

Display file directory

Kill file (and crea*e space)

Window split and display two files

Copy ile

Rename file

CURSOR CONTROL

By character

By word

By line

By sentence

By paragraph

By block

By screen

To beginning or end of workspace

To beginning or end of document

Horizontal scroll

Vertical scroll

Continuous vertical scroll

DELETE

By character

By word

By line

To end of line

By sentence

To end of sentence

By paragraph

By block

Continuous

INSERT

Glossary

Typeover

Insert mode

Push ahead

Split and glue a line

Line buffer

Copy and move paragraph

Copy and move region

Delete and restore

SEARCH

Find phrase anywhere

Find with user replace

Find and replace n times

Find and replace all in document

Use wildcards

Ignore upper/lower case

Place markers

SCREEN FORMAT

Format entire text

Format different parts

Set line length

Set tabs with cursor

Set tabs by command

Paragraph tab

Margin control

Page height control

Automatic capitalization

Automatic lowercase

Transpose letters and words

TEXT FORMATTER

Display as printed

Justification control

Word wrap on/off

Print control display on/off

Print file while editing another

Mail-merge or file-merge

Most printers supported

Line space control

LAYOUT

From menu

Menu may be skipped

Width limitation

PAGE CONTROL

One-line heading

Multi-line heading

Heading and footing

Page numbering

Omit age number

Odd/even page distinction

Conditional new page

TEXT CONTROL

Justify

Centre

Phantom hyphen

Multiple columns

Reverse line feed

PRINTER CONTROL

Underline

Boldfrce

Vary boldface intensity

Super- and subscript

Change ribbon colours

Change control characters

Proportional spacing

Alternate pitch

Non break space

Overprint next character

Strike out

User printer controls

Character width (daisy wheel)

Microjustify on/off

Number of printer strings

Leng of printer string

Printer drivers supported

OUTPUT CONTROL

Interrupt, resume

Alter format

Pause for text entry

Pause for variable entry

Start/stop as designated

Print multiple documents

Print multiple copies

ADDITIONAL FEATURES

Form letters

Dictionary

Document index

Arithmetic capabilities

Grammar checker

Ability to run program from within

Labels

Uses ASCII text files

SPREADSHEET

GENERAL

Memory requirements

Screensize (columns)

Product disk resident

Software protected

Macro facility

WORKSHEET

Absolute size (rows X cols)

Smallest number

Largest number

Number of decimal places

Maximum text cell size

Defaults:

- Column width
- Numeric overflow
- Numeric alignment
- Text alignment
- Automatic recalculation

Delineating types of input

- Value
- Formula
- Text

Protected cells

Hidden cells

Cell addressing by cell or name

COMMANDS

- Blanking
- Calculation order
    - Rows L to R, T to B
    - Cols T to B, L to R
    - Automatic recalc on/off
- Copying/replication
    - row(s)
    - column(s)
    - blocks
    - values
    - formula adjustment
    - relative adjustment
- Deleting
    - all
    - row
    - column ,
    - block
    - entry
- Editing
    - addressable
    - insert
    - change
    - delete
    - formula adjustment

File handling

- initialize diskette
- erase a file

Formatting

- entry
- row(s)
- column(s)
- all
- column width (single or range)
- dollars/cents
- integer
- decimal precision
- scientific notation
- numeric alignment L/R
- text alignment L/R
- text beyond cell
- graph

Inserting

- row(s)
- columns(s)
- formula adjustment

Loading/Merging

- file directory
- part of worksheet
- into specified cells
- formula adjustment
- consolidation

- Moving
    - row(s)
    - column(s)
    - formula adjustment
- Printing
    - formula display
    - part of worksheet
    - page specifications
    - printer specifications
    - pause between pages
    - headers and footers
    - formulas and formats
    - to diskette
- Saving
    - part of worksheet
    - text and values only
    - file directory
    - file name on screen
- Sorting
    - number of fields
- Title Locking
    - horizontal
    - vertical
    - both
- Windowing
    - horizontal
    - vertical

synchronized

TECHNICAL DETAILS

Operating system

Multi user capability

Hard disk compatible

Language system is written in

User supplied language required

Terminal setup

Menu driven terminal table

Menu driven printer table

Installation with entry of control codes

MATHEMATICAL FUNCTIONS

Boolian logic

Absolute value

Internal rate of return

Net present value

Square roots

Trigonometric functions

Linear regression

Minimum/Max value

Average

Lookup tables

Choose

Count

Exponential

Integer

Pi

Sum

DATABASE

CAPABILITIES

Max no of characters per field

Max no of fields per record

Max no of characters per record

Max no of records per file

Max no of files open simultaneously

Max no of key fields

Max length of key fields

Max no of disk drives

GENERAL

Menu driven

Linked files

Terminal customization

Printer customization

Installation of printer control codes

Availability of programming constructs

User defined data entry screens

Help command

Data ecurity and passwords

Command file

So t function

Edit mask

DATA FIELD DETAILS

Field naming

Field data types supported

Validation types

Multi-field keys

Modification allowed

FILE MAINTENANCE

Updates one record at a time

Multi-record updates

System triggered updates

Global updates

'Audit trail

SELECTION

Conditions on one field

Conditions on multiple fields

Complex nestec conditions

Maximum number of conditions' allowed

Storage of selection criteria

REPORT GENERATOR

Predefined format

Maximum number of predefined formats

Multiple input files

Title

Heading

Footing

Field naming

Field data types supported

Validation types

Multi-field keys

Modification allowed

FILE MAINTENANCE

Updates one record at a time

Multi-record updates

System triggered updates

Global updates

'Audit trail

SELECTION

Conditions on one field

Conditions on multiple fields

Complex nested conditions

Maximum number of conditions allowed

Storage of selection criteria

REPORT GENERATOR

Predefined format

Maximum number of predefined formats

Multiple input files

Title

Heading

Footing

Page numbering

Column width specification

Text inserts

Margins

Lines between records

Summary breakpoints

Sums

Means

Minimums

Maximums

Roots

Statisics

Average

Record numbering

Selection process

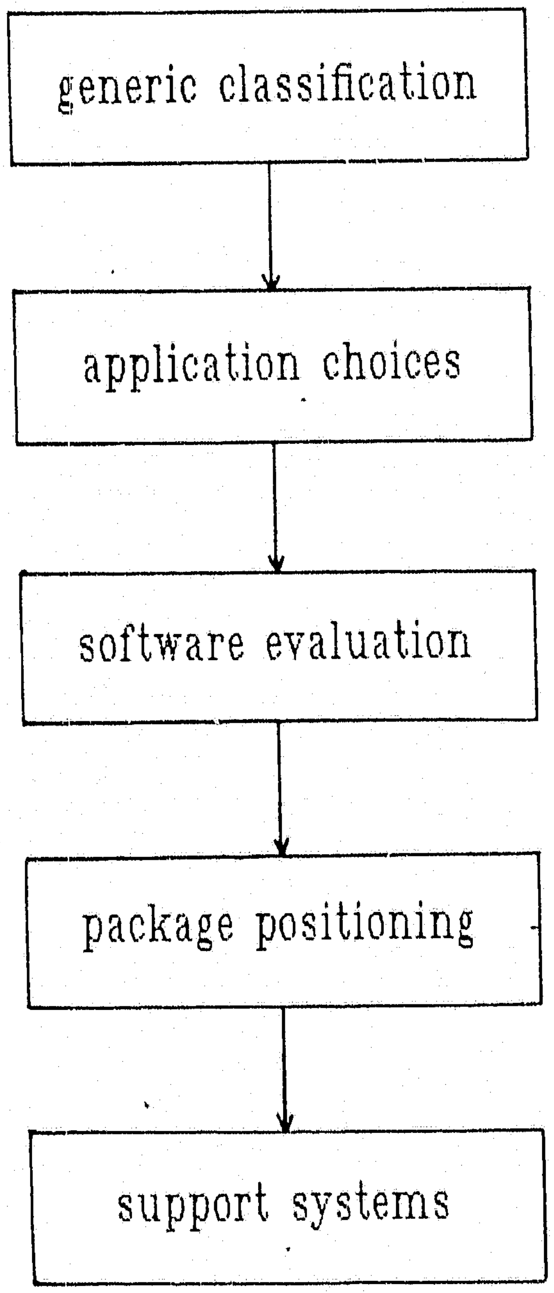

Fig. 1

GROUP	APPLICATION	SYSTEM DESIGN	DOCUMENTATION	EDUCATION	INSTALLATION	ONGOING SUPPORT	USER LEVEL	PRICING
A	Games, Hobby, some Utilities	Bug free (presumably)	In English Well written	N/A	N/A	N/A	Enthusiast	Very low
B	Simple package	Bug free Proven Top quality Reputable	Non technical Well written Quality presentation	Optional	Not required (cash & carry)	Required (hotline)	High	Low
C	Complex package	Bug free Proven Top quality Reputable	Non technical Well written Quality presentation	Optional (depending on user)	Required	Required	All	Low-Med educat. install extra
D	Complex system	Bug free Proven Reputable	Non technical Well written Quality presentation	Required	Required	Required	All	High incl. install & educ.
E	Application generator	Secure (presumably)	Operator manual	Required	Required	Required	All	High +
F	Custom software	Secure (presumably)	Operator manual	Required	Required	Required	All	Very high

Generic Classification Table

Fig. 2

*support systems

o ensure application skills

o construct support matrix:

AA/BB - financial modelling

AA/BB - business graphics

BB/AA - data base

BB/AA - filing systems

CC - stock market

AA/BB - wordprocessing

AA/BB - mailing

DD/AA - accounting

DD/AA - general business

AA - utilities

AA - languages

Fig. 3

REFERENCES

REFERENCES

1) A. L. Taylor III, 'The Software Hard Sell', TIME magazine, June 20, 1983, p.52.

2) Hal Kenny, Director of 'Future Computing', USA market research company. The above information was obtained during a talk given by Kenny at the Carlton Hotel, October 1984.

3) S. M. Murtha and M. Waite, CP/M Primer, p.11.

4) D. Wise and C. Harris, 'Apple's New Crusade: a bold new plan to take on IBM in the office', BUSINESSWEEK, November 26, 1984, p.66.

5) A. L. Taylor III, 'The Software Hard Sell', TIME magazine, June 20, 1983, p.52.

6) A. L. Taylor III, 'The Wizard Inside the Machines', TIME magazine, April 16, 1984, p.46.

7) Ibid.

8) Ibid, p.48.

9) A. L. Taylor III, 'The Software Hard Sell', TIME magazine, June 20, 1983, p.52.

10) S. Alsop (ed), INC, January 1982, p.66.

11) ComputerWeek, August 20, 1984, p.12.

12) J. Markoff and P. Freidburger, 'The Microsoft success story', Computing S.A , September 23, 1983, p.11.
(Until otherwise indicated, all information pertaining to Microsoft has been obtained from the above source.)

13) M. Moritz, 'A Hard-Core Technoid', TIME, April 16, 1984, p.59.

14) Ibid.

15) Until otherwise indicated, all information pertaining to VisiCorp has been obtained from telexed information on 'VisiCorp Corporate Background', Telex no' 172159, 2895 Zanker Road, San Jose, CA 95134, August 1983, pp 1 - 6.

16) D. Caruso, 'In US software's a poker game: strategies boomerang', Computing S.A , April 13, 1984, p.10.

17) See footnote no' 15.

18) ComputerWeek, August 20, 1984, p.12.

19) 'The voice behind VisiOn', an interview with Dan Fylstra, Chairman of VisiCalc, PC User, January 1984, p.58.

20) ComputerWeek, August 20, 1984, p. 12.

21) S. Alsop II, 'Software Arts Wrote the First Best-Seller', INC, January 1982, pp.71-75.
(Until otherwise indicated, all information pertaining to 'Software Arts', has been obtained from the above source.)

22) Computerweek, August 20, 1984, p. 12.

23) See footnote no,' 21.

24) S. Alsop II (executive editor), INC, January 1982, p.68.

25) D.Caruso, 'In USA software's a poker game', Computing S.A , April 13, 1984, p.23.

26) S. Alsop II (executive editor), INC, January 1982, p.68.

27) Ibid.

28) 'Software News', ComputerWeek, 17 September, 1984, p.20.

29) D. Caruso, 'In USA software's a poker game', Computing S.A , April 13, 1984, p.10.

30) Ibid.

31) Ibid.

32) A. L. Taylor III, 'The Wizard inside the Machines', TIME, April 16, 1984, p.49.

33) D. Caruso, 'IN USA software's a poker game', Computing S.A , April 13, 1984, p.11.

34) Ibid.

35) A. L. Taylor III, 'The Wizard inside the Machines', TIME, April 16, 1984, p.49.

36) S. Alsop II (executive editor), INC, January 1982, p.68.

37) A. L. Taylor III, 'The Wizard inside the Machines', TIME, April 13, 1984, p.49.

38) Ibid.

39) D. Caruso, 'In USA software's a poker game', Computing, April 13, 1984, p.11.

40) Ibid.

41) Ibid.

42) A. L. Taylor III, 'The Software Hard Sell', TIME, June 20, 1983, p.53.

43) 'Big league boys are climbing in', ComputerWeek, 30 April 1984.

44) Ibid.

45) Ibid.

46) A. Vaughan, 'IBM moves in on PC software marketplace', ComputerWeek, 4th June 1984.

47) A. L. Taylor III, 'The Software Hard Sell', TIME, June 20, 1983, p.52.

48) Ibid, p.53.

49) Ibid.

50) Hal Kenny, Director of 'Future Computing', USA market research company. The above information was obtained during a talk given by Kenny at the Carlton Hotel, October 1984.

51) Ibid.

52) 'Software Retailing: Quick Bucks - and Quicksand', Marketing 1984.

53) A. L. Taylor III, 'The Software Hard Sell', TIME, June 20, 1983, p.52.

54) P., E. De Witt, 'The Stepchild Comes of Age', TIME, March 5, 1984, p.55.

55) W. A. Freyenfeld, 'Telesoftware', Tne National Computing Centre Limited, 1983, pp.5 - 49.

56) 'So You Wanna Be a Software Star - A Look at Alternative Distribution', The Computer Entrepreneur, 1984, p.3.

57) Ibid.

58) Ibid.

59) W. A. Freyenfeld, 'Telesoftware', The National Computing Centre Limited, 1983. pp.5 - 49.

60) A. Vaughan, 'Computer programs on British airwaves', ComputerWeek, 20 February 1984, p.4.

61) 'Software going free - but making some rich', Computing S.A, April 13, 1984, p.13.

62) Ibid.