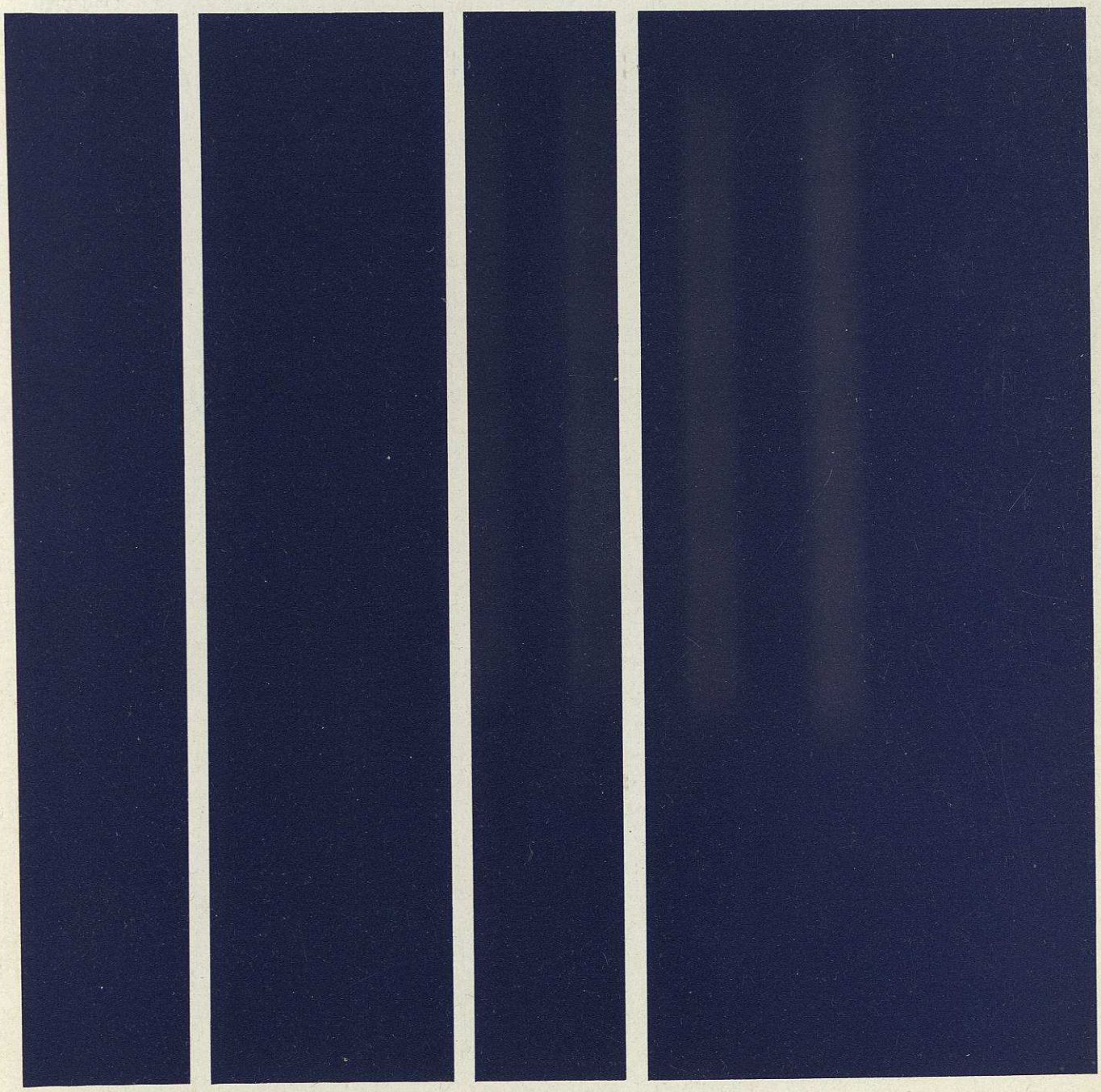
Report Series
No 22

Applications Packages

January 1981



The Butler Cox Foundation

Abstract

Report Series No 22

Applications Packages

by Martin Ray

January 1981

During the past several years, there has been a renewed interest in the use of applications packages, and the prevailing unfavourable economic climate has undoubtedly been partly responsible for this. But underlying trends in systems technology and its economics are also at the root of this revival in interest in the use of applications packages. Of these trends, the most important is the increasing cost of developing applications systems, both in absolute terms, and more particularly, in terms relative to the cost of computer hardware.

The main purpose of this report is to inform those who manage an organisation's information systems function about the issues concerned with the use of applications packages. The report reviews both the overall shape of, and the trends in, the applications packages marketplace, and it contrasts the approaches that eight major suppliers of applications packages take in designing and supporting their products. The report also analyses the experience that many Foundation members have of selecting and using applications packages.

The findings of our research of the marketplace, of the suppliers and their products, and of the experience that members have of using packages are brought together to provide advice at both the strategic level and the tactical level. The report provides guidance at the strategic level about the role that applications packages can now play in an organisation's overall approach to systems development, and it identifies the implications that this role has for those who manage the information systems function. The report provides guidance at the tactical level on selecting and purchasing applications packages.

The Butler Cox Foundation is a research group that examines major developments in the fields of computers, telecommunications and office automation on behalf of its subscribing members. The Foundation provides a set of "eyes and ears" on the world for the systems departments of some of Europe's largest organisations.

The Foundation collects its information through its office in London and also through its associated offices in Europe and the US. It transmits its findings to its members in three main ways:

- Through regular *written reports* that give detailed findings and substantiating evidence.

- Through *management conferences* for management services directors and their senior colleagues, where the emphasis is on the policy implications of the subjects studied.

- Through *professional and technical seminars* where the members' own specialist managers and technicians meet with the Foundation research teams to review their findings in depth.

The Foundation is controlled by a Management Board whose members include representatives from the Foundation member organisations. The responsibilities of the Management Board include selecting topics for research and approving the Foundation's annual report and accounts, which show how the subscribed research funds have been employed.

Report Series No 22

APPLICATIONS PACKAGES

by Martin Ray

January 1981

Butler Cox & Partners Limited Morley House 26 Holborn Viaduct London EC1A 2BP

CONTENTS

CHAPTER 1

INTRODUCTION

Applications packages have been marketed and used since the mid-1960s, but there was a period during the 1970s when many organisations did not favour using them. Now, however, there is a renewed interest in the use of applications packages, and this is illustrated by the way the number of packages sold has increased rapidly during the past few years. This renewed interest has come about for the following reasons:

1. Economies of scale
   Computer equipment has become much cheaper because of advances in both the design of the equipment and the manufacturing processes. There have been no comparable advances in either the design or the manufacture of software. The production of standard software, with its development costs spread over many users, is one method of achieving software cost reductions equivalent to equipment cost reductions. The trend to standard software has, for some time, been evident in operating systems and environmental software such as database management systems. It is now also apparent in applications software.

2. Increased demand for systems
   The demand for computer-based systems continues to grow in almost all organisations. This increased demand applies not only to commercial data processing applications, but also to new applications in the office, in the factory and in the public domain. Systems development staff are in short supply, however, and this situation will almost certainly not change in the medium-term future. It makes sense, therefore, for suppliers to concentrate some of this scarce resource on developing standard systems that many organisations can then use. Individual organisations can then concentrate the effort of their own systems development resource on those systems activities that are unique to their own business.

3. Reliability of software
   The low level of software reliability continues to be a problem that manifests itself in several ways, including delays to development projects and the high cost of maintaining operational systems. The use of improved and more scientific methods of developing software is currently the focus of much attention but, in the meantime, organisations need to use pragmatic methods to reduce the impact of the low level of software reliability. One method is to use program code that has previously been used in other installations and has become more reliable as a result.

4. Increased market for packages
   There is now a larger market for applications packages than there was a decade or more ago. This larger market has developed as a result of both the increase in the number of computers installed and the general increase in computer power available. In chapter 3 (on page16) we report that our research has shown that it takes between five and fifteen times more effort to develop and market a package than it does to develop a once-off bespoke system. This means then that before a company can justify the investment that is essential to properly design, market and support packages it needs to be assured that a large potential market exists for those packages.

5. Changed marketing strategy
   The continuing reduction in the cost of computer hardware has caused equipment suppliers

to reappraise their marketing strategies. The first evidence of this reappraisal came from the minicomputer OEMs in the mid-1970s. They found that, to sell equipment valued at between £30,000 and £50,000, they had to offer packaged applications software as well, in order to keep the total cost to the purchaser to a reasonable figure.

Over the past few years, major system suppliers have also found it necessary to review the strategic importance of applications software. They are beginning to see applications software not just as a necessary precondition for selling hardware, but also as a source of added revenue that is, and will continue to be, of increasing commercial importance to them. This trend towards the increasing commercial importance of applications software is highlighted by the importance that all hardware suppliers are now placing on office automation. Many computer-based applications in the office (of which word processing is an obvious example) offer suppliers the opportunity to sell a combination of hardware and a standard applications package.

THE PURPOSE AND THE STRUCTURE OF THIS REPORT

The main purpose of this report is to advise those who manage the information systems function about the strategic and the tactical issues concerned with applications packages. The report reviews the major trends in the applications packages marketplace and considers the impact of applications packages both on systems development and on systems strategy as a whole. At a more detailed level, the report provides advice on how to evaluate and purchase applications packages.

Chapter 2 reviews the overall shape and trends of the applications packages marketplace. Chapter 3 examines the product development strategies and the marketing strategies of several major applications package suppliers, and identifies common and different trends in those suppliers' operations.

Chapter 4 reports the findings of both a survey of, and several interviews with, Foundation members, which we conducted as part of the research for this report.

Chapter 5 brings together the findings of our research concerning the selection, purchase and use of applications packages. Finally, chapter 6 examines those aspects of overall systems strategy that applications packages may have an effect on.

THE SCOPE OF THE REPORT

This report is concerned with a limited part of all of the proprietary software that is available both from computer suppliers and the independent software suppliers. Proprietary software as a whole can be classified broadly in the following ways:

- Environmental software such as operating systems, teleprocessing monitors, database management systems and networking software.
- Implementation software tools such as computer languages, debugging and testing aids, and query languages.
- Utility software such as library routines and sort/merge packages.
- Operational software tools for measuring performance and scheduling computer tasks.
- Applications software packages for use in functional data processing systems, in specific industries and in business communications systems.

In practice, the distinction between applications packages and some of the other types of proprietary software is blurred. For example, several applications packages include a high-level report-generation language, and some applications packages use a database management system for file management tasks.

We have concentrated on one aspect of the applications packages marketplace because we know that that aspect is of universal interest to Foundation members. We refer to the market for general data processing applications packages, which includes packages for functions such as finance, production and personnel. We have excluded packages that are intended to be used only in a specific industry (such as banking). We have also excluded packages that are concerned with business communications (such as word processing, electronic mail and so on). Much of this report is, however, also directly relevant to both those types of packages.

INTENDED READERSHIP

This report is written mainly for those who manage the information systems function and all those who are concerned with systems development. It is also intended to be read by (and to be helpful to) those data processing staff who may be involved in selecting and evaluating applications packages.

The report is also intended to be of value to management and executive staff outside of the data processing area who are concerned with purchasing applications software. Finally, and by implication, it is intended to be of value to the management of those organisations that are concerned with the design, development and marketing of applications software.

CHAPTER 2

THE APPLICATIONS PACKAGES MARKETPLACE

The economics of commercial data processing have, for the past few years, been dominated by the following three factors:

- The price of hardware has declined as a result of improved production methods and greater competition.
- The cost of producing software has risen at approximately the rate of salary inflation.
- The shortage of programmers and analysts is chronic, particularly in relation to the rate at which the demand for computer-based systems is growing.

These three factors combine to create difficulties both for those who manage the information systems function and for the suppliers of hardware and software. The manager of the information systems function is frustrated in his endeavour to harness cheap computer power to meet the needs of his organisation. The hardware supplier sees his profit margins eroded as the cost of a given level of computing power decreases. Many hardware suppliers now supply, and charge for, software in addition to hardware, as a way of adding value to their sales. The independent software supplier needs to devise and sell the same standard software to many users. By doing this he can both reduce the unit cost to each purchaser and finance properly the design, marketing and support of his products.

It is against this background that we now examine the main trends in the applications software marketplace. We begin with a review of the overall trends, which are independent of any particular national marketplace. We then analyse in some detail both the structure of the marketplace in the United Kingdom and the products available in that marketplace. In carrying out the various analyses of the United Kingdom marketplace, we have not made reference to specific suppliers or products. We believe, therefore, that our findings in the United Kingdom will, in general terms, be valid also for other national markets.

THE OVERALL TRENDS IN THE MARKETPLACE

Most of the available statistics concerning the size and the rate of growth of the applications packages marketplace relate to the United States. Consequently, our discussion of the overall trends refers primarily to that country. We believe, however, that the overall trends in the United States are similar to those elsewhere in the Western World, even though the size of the software marketplace in the United States is about three times as large as the size of the equivalent market in Europe. We believe, therefore, that our discussion of the overall trends in the applications packages marketplace is valid not only for the United States but also for Europe.

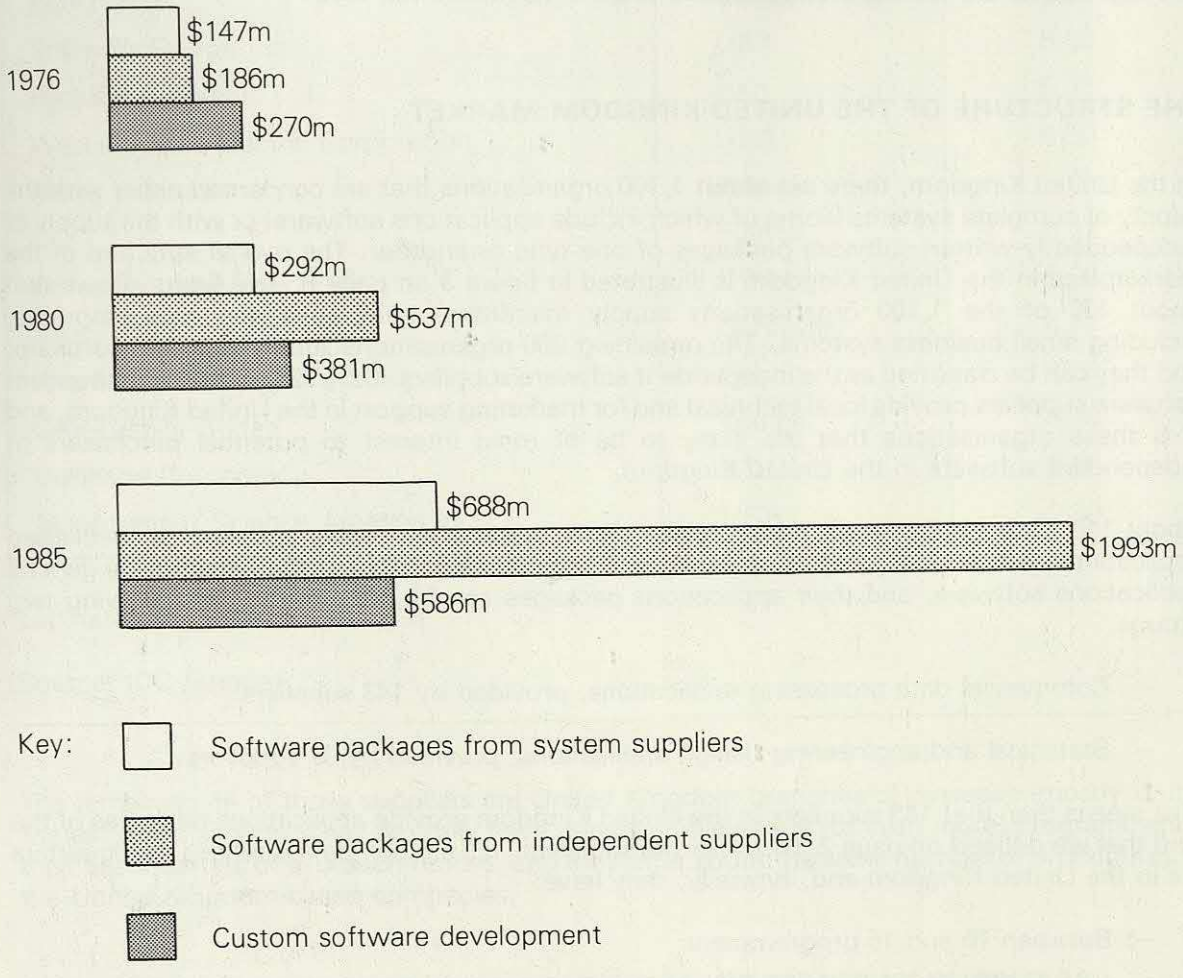
Before we examine the trends in the market for applications packages, we first need to consider the wider field of software packages (which includes not only applications software, but also operating systems, database management systems, query languages, etc.). The suppliers of software packages can be divided conveniently into the two following categories:

- System suppliers, such as IBM, ICL, Honeywell, etc. Several independent sources (such as International Data Corporation) have estimated that, in the United States, system suppliers account for about half of the total software market.

- Independent software vendors, such as Informatics, MSA, Cincom, Pansophic, etc. Most of the published statistics indicate that the market share of this category of suppliers is growing at a faster rate than the market share of the system suppliers. Figure 1 shows that independent software suppliers accounted for less than half of the market in the mid-1970s, but that they are likely to supply nearly two-thirds of the market by 1985. The figure shows also that the total software market is expected to increase by 170 per cent in the five years up to 1985, although the market for custom-made software will increase at a much slower rate than will the market for standard software packages.

It is not easy to find statistics that break down the revenues from all types of software packages into the constituent sectors of the software market that we identify on page 2. What is clear

Figure 1 Software sales in the United States



(Source: M. Phister)

though is that, in 1978, the market share and the annual growth rate of the three main sectors of the independent software market in the United States were as follows:

- Systems software accounted for 15 per cent of the market, and the volume of sales was growing at 21 per cent per annum.
- Utility software accounted for 36 per cent of the market, and the volume of sales was growing at 36 per cent per annum.
- Applications software accounted for 48 per cent of the market, and the volume of sales was growing at 39 per cent per annum.

The European software market, measured by the revenue earned by software companies, is dominated largely by United States companies, although the local software houses both in France and the United Kingdom provide effective competition. Figure 2 shows the revenue earned in Europe by independent software suppliers in 1979. The revenue amounts in figure 2 are taken from a recent report published by International Data Corporation, and the report also estimates that the total value of the United Kingdom packaged software market in 1979 was £33 million. The report also predicts that the market will grow by 25 per cent per annum up to 1983, when the total revenue will be £82 million for that year. The value of the total European packaged software market was estimated to be £165 million for 1979.

THE STRUCTURE OF THE UNITED KINGDOM MARKET

In the United Kingdom, there are about 1,100 organisations that are concerned either with the supply of complete systems (some of which include applications software) or with the supply of independently-written software packages of one type or another. The overall structure of the marketplace in the United Kingdom is illustrated in figure 3 on page 8. The figure shows that about 130 of the 1,100 organisations supply mainframe computers and minicomputers, including small business systems. The remaining 950 organisations supply standard software, and they can be classified as the independent software suppliers. Only 325 of those independent software suppliers provide local technical and/or marketing support in the United Kingdom, and it is these organisations that are likely to be of most interest to potential purchasers of independent software in the United Kingdom.

About 145 of the 325 organisations provide either systems and utility software or specialised applications software for particular industries. The remaining 180 organisations provide general applications software, and their applications packages can be divided into the following two groups:

- Commercial data processing applications, provided by 143 suppliers.
- Statistical and engineering design applications, provided by 37 suppliers.

This means then that 143 suppliers in the United Kingdom provide applications packages of the kind that we defined on page 2. Of those 143 suppliers, 129 are companies whose headquarters are in the United Kingdom and, typically, they have:

- Between 10 and 15 programmers.
- Between 3 and 5 marketing staff.
- Between 3 and 10 products.

Figure 2 Packaged software revenues in Europe in 1971

Supplier	Country of origin	$ (millions)
Cincom Systems Inc.	USA	12.50
Computer Associates Inc.	USA	9.35
Informatics Inc.	USA	9.30
Software AG	Germany	8.90
Applied Data Research Inc.	USA	8.15
Altergo Software Ltd.	UK	7.52
Hoskyns Group Ltd.	UK	7.00
Pansophic Systems Inc.	USA	6.80
Peterborough Data Processing Services Ltd.	UK	6.50*
ADV/Orga	Germany	6.36
Software Design Inc.	USA	5.90
Package Programs Ltd.	UK	5.77
Westinghouse Electric Corporation	USA	5.50
Metra International	France	5.30
CGI	France	5.22
Arthur Anderson and Co.	USA	5.00
SAP	Germany	4.71
The European Software Company Ltd.	Eire	4.00
CAP-Gemini-Sogeti	France	3.40
Cullinane Corporation	USA	3.26
Management Science America Inc.	USA	2.50

*Estimated

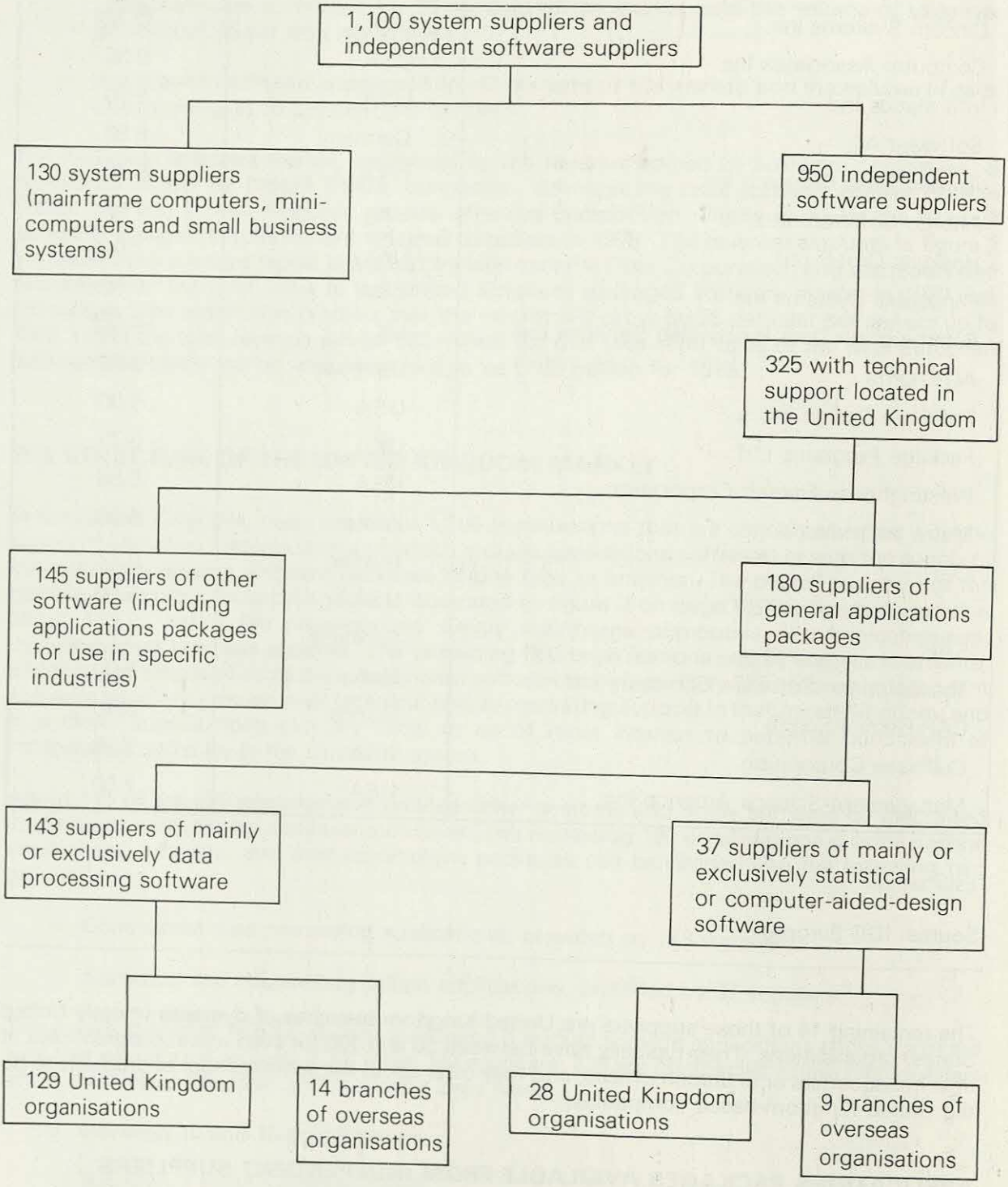
(Source: IDC Europe)

The remaining 14 of those suppliers are United Kingdom branches of overseas (mostly United States) organisations. They typically have between 50 and 100 (or even more) programmers at their headquarters or in branch offices, and they offer about the same number of products as do the United Kingdom-based companies.

APPLICATIONS PACKAGES AVAILABLE FROM INDEPENDENT SUPPLIERS

As part of the research for this report we undertook an analysis of the applications packages available in the United Kingdom market from independent sources. (By independent sources we

Figure 3 The structure of the United Kingdom applications packages market at the end of 1980



mean those sources that do not require the potential user either to purchase computer equipment or to use a bureau in order to make use of the applications software.) Our analysis showed that the number of software packages on offer to users can be categorised as follows:

- 292 system software packages.
- 637 utility software packages.
- 208 applications packages for specific industries.
- 1,200 general data processing applications packages.

We analysed in detail the types of applications packages available for general data processing applications, and the results of the analysis are shown in figure 4. The figure shows that 39 per cent of the packages are for use in financial applications, with administration applications and production planning (or production control) applications each catered for by about 18 per cent of the packages.

Figure 4 Types of applications packages available in 1980 in the United Kingdom

Type of application	Number of packages	Typical applications
Financial accounting	470	Nominal ledger, purchase ledger, sales ledger
Production planning and control	240	Bills of materials, inventory control, production control, purchasing
Administration	210	Asset accounting, financial planning, job costing, project management, telephone accounting
Employee administration	120	Payroll, personnel records, pension records
Sales and marketing	100	Sales order processing, invoicing, mailing, market research
Distribution	40	Route planning, fleet maintenance

In figure 5, we show the growth in the number of applications packages available in the United Kingdom from independent suppliers. Each entry in figure 5 shows the proportion of packages available today that were brought to the market in each of the time periods shown in the figure. From the figure, the following three main trends concerning the growth in the availability of packages can be seen:

- More than half of the packages available today first came onto the market between 1977 and 1979.

- At the end of the 1970s, packages for use on minicomputers accounted for more than half of all the packages available, whilst packages available for use on mainframe computers accounted for only one-third of the total.

- The number of new mainframe packages coming onto the market reached a peak in the mid-1970s, and has declined since then.

Although figure 5 does not illustrate it, the rate of growth in the number of microcomputer packages was, by 1979, approximately twice that of mainframe packages. The total number of microcomputer packages available is more or less doubling each year now, and, if this trend continues, microcomputer packages will dominate the packages marketplace in the fairly near future.

Figure 5 Percentage of applications packages available in the United Kingdom at the end of the 1970s, tabulated by the periods in which they were introduced

Type of computer	Prior to 1970	1970 to 1973	1974 to 1976	1977 to 1979	Total
Mainframe	3.8	6.5	12.9	10.7	33.9
Minicomputer	1.1	5.4	15.2	34.7	56.4
Microcomputer	–	–	0.2	9.5	9.7
Total	4.9	11.9	28.3	54.9	100.0

Figure 6 shows, for each type of computer and for each of the most common type of application, the number of packages available in the United Kingdom prior to 1975, and the number that became available in the period 1975 to 1979. From the analysis shown in figure 6 we conclude that:

- Minicomputer packages for financial applications have been the fastest growing market sector in the period 1975 to 1979.

- Mainframe and minicomputer packages, both for production planning (or production control) applications and for employee-administration applications, have been areas of high growth.

- The range of microcomputer packages now available is widest for those application areas for which there is already a wide choice of mainframe or minicomputer packages.

THE USAGE OF APPLICATIONS PACKAGES

Our interviews with suppliers of packaged software (which we report on in chapter 3) showed us that the software industry commonly expects the effort required to develop a package to be

between five and fifteen times the effort required to develop an equivalent bespoke system. With this expectation in mind, we analysed the number of worldwide purchasers of the packages available in the United Kingdom from independent suppliers for three selected applications — payroll, general ledger and financial planning.

Figure 6 Number of applications packages available in the United Kingdom, tabulated by type of application and type of computer

Type of application	Type of computer				
	Mainframe		Minicomputer		Micro-computer
	Pre 1974	1975-79	Pre 1974	1975-79	1975-1979
Financial accounting	49	46	58	228	50
Administration	42	48	15	70	12
Sales and marketing	11	11	4	55	12
Employee administration	13	29	25	59	15
Project management	17	24	9	27	7
Production planning and control	33	46	24	97	19
Distribution	1	10	5	23	4
Statistics	14	23	9	9	10

We examined a total of 265 packages, and the results of our analysis are shown in figure 7. They show that 128 packages (or about one-half of all those we examined) had each been purchased by fewer than five organisations worldwide. Our survey of, and our interviews with, Foundation members lead us to conclude that few potential purchasers would regard packages that have such a small customer base as being worthy of detailed evaluation. In addition, it is most unlikely that a package that has such a small customer base could be commercially successful.

The results of our analysis indicate also that 86 packages (or about one-third of all those we examined) had each been purchased by 16 or more organisations worldwide. This size of customer base would lead most potential purchasers to regard those packages as being worthy of further evaluation. Also, packages that have this size of customer base can reasonably expect to achieve commercial success, and this commercial success will enable the suppliers concerned to provide the level of support that purchasers require.

Of the 265 packages we examined, only 30 have a customer base of 100 or more. Packages such as those are likely to be able to cater for a wide variety of end-user requirements. They are likely also to warrant not only effective marketing and technical organisations to support them, but also a comprehensive programme of product development. Of the 30 packages that fall into this

category, 25 originate in the United States, and only 4 originate in the United Kingdom. (The other one originates in Canada.)

Figure 7 Worldwide purchasers of selected packages available in the United Kingdom as at the end of 1980

Application	Number of packages available	Number of packages with 1 to 4 purchasers	Number of packages with 5 to 15 purchasers	Number of packages with 16 to 100 purchasers	Number of packages with more than 100 purchasers
Payroll	103	56	16	18	13
General ledger	94	40	20	22	12
Financial planning	68	32	15	16	5
Total	265	128	51	56	30
Percentage of total	100	49	19	20	12

SUMMARY

In this chapter we have analysed briefly the structure of the marketplace for applications packages. Our analysis has shown that the number of new applications packages available for use on mainframe computers is now growing slowly, whereas, for the past five years, the majority of new packages have been designed for minicomputers. However, the number of packages available for use on microcomputers is now growing at an explosive rate, and if this trend continues, microcomputer packages will soon dominate the applications packages market.

At first sight, the applications packages market is served by several hundred organisations, each offering several products. Many of those products, however, have been purchased by only a few organisations, and it is doubtful if they can really be classified as packages. Our view is that this proliferation of products that only a few potential users purchase is a sign of an immature marketplace. Consequently, we expect that the number of suppliers and products will diminish eventually. We expect, then, that the surviving products and suppliers will be truly able to meet the requirements of a diverse end-user community.

CHAPTER 3

THE SUPPLIERS OF APPLICATIONS PACKAGES AND THEIR PRODUCTS

In the previous chapter of this report we indicated that only a few of the suppliers in the United Kingdom applications packages marketplace have a significant number of customers for any one of their products. We said, moreover, that only those few suppliers are likely to succeed in the applications packages marketplace.

We decided, as part of our research for this report, to seek the assistance of nine suppliers that we selected as being (or as promising to be) successful in the applications packages marketplace. We concentrated on three categories of suppliers of packages that are designed to be used with mainframe computers and minicomputers, as set out below:

- System suppliers. We invited IBM (United Kingdom) Limited, International Computers Limited, and Digital Equipment Company Limited to participate in the research, and all three agreed to do so.

- Computer services companies. All three of the computer services companies (Hoskyns Group Limited, Dataskil Limited and RTZ Computer Services Limited) that we invited to participate in the research agreed to do so.

- Independent package suppliers. We invited three independent package suppliers to participate in the research. Two of them (Management Science America Limited and Package Programs Limited) agreed to participate, but Peterborough Data Processing Services Limited declined our invitation. (Management Science America Limited is the United Kingdom subsidiary of Management Science America Inc.)

All of those eight supplier organisations provided us with valuable assistance with our research. We conducted an in-depth interview with a senior staff representative from each organisation. The aim of each interview was to allow us to gain a thorough understanding of the supplier's policy, approach and product range, and also to identify any similarities or differences there are between the three different categories of suppliers of applications packages.

In addition, we examined one of the products supplied by each of five of the participating suppliers both for technical quality and level of documentation. The five products were:

- The MAS range of packages provided by Hoskyns Group Limited.

- The Prosper system provided by Dataskil Limited.

- The FASCIA package provided by RTZ Computer Services Limited.

- The Q-PAC package provided by Management Science America Limited.

- The MMS range of systems provided by Package Programs Limited.

The approach that we have taken in this chapter in presenting the findings of our research with

package suppliers is to compare and contrast the approaches taken by the participating suppliers under the six following headings:

- The customers for applications packages.
- The achievement of customer satisfaction.
- The range of hardware supported by packages.
- The relationship suppliers have with hardware manufacturers.
- The applications areas selected by the suppliers.
- The origins and the development of packages.

Although the findings reported in this chapter relate specifically to the eight organisations that participated in the research, the conclusions that we have come to are, in our view, likely to apply to most, if not all, suppliers of applications packages.

THE CUSTOMERS FOR APPLICATIONS PACKAGES

Most applications packages are now sold direct to end users within an organisation, rather than to the organisation's data processing function. The suppliers of packages, and in particular the independent suppliers and the computer services companies, have been forced to sell direct to end users because of the hostile attitudes some data processing professionals take towards applications packages. However, most of the suppliers that we interviewed told us that the number of enquiries they receive from data processing departments is now increasing. They added that this increase seems to indicate that data processing professionals have had a change of attitude towards applications packages. Our own research both of user experience of, and attitudes to, applications packages, which we discuss in chapter 4 of this report, confirms that the attitudes of data processing professionals towards the use of applications packages have recently undergone a change.

Of the three categories of package suppliers, the system suppliers have the closest relationship with an organisation's data processing department. On the other hand, the independent suppliers have closer ties with end users, and often those suppliers deal with the data processing department in an organisation only at the time when a package is installed or when in-house modifications to a package are being carried out.

Most of the suppliers of packages that participated in the research encourage the formation of user groups. In general, organised user-group liaison with the client base is valued more by those suppliers whose total revenue depends largely on providing applications packages than it is by those suppliers whose revenue from applications packages forms only a small part of their total revenue. Applications packages user groups are usually controlled by the customers, and the supplier will attend a user-group meeting only if it is invited to do so. From a supplier's point of view, one of the most useful functions that an applications package user group fulfils is as a forum for determining the priorities for new facilities to be included in future releases of the package.

THE ACHIEVEMENT OF CUSTOMER SATISFACTION

Applications packages can be described as generalised solutions to requirements that most, if not all, potential customers perceive as being unique to their own particular circumstances. For

a supplier, an important aspect of both achieving a sale and ensuring customer satisfaction with an installed package is either to eliminate or to disguise this apparent contradiction. In order to solve this problem, package suppliers can take one of the following two contrasting approaches:

- The service-oriented approach, which provides a flexible service by customising the package to meet each customer's particular requirements.

- The product-oriented approach, which provides a flexible product that can be tuned to satisfy each customer's requirements without modifying either the concepts that underlie the package or the coding itself.

As we discuss below, the system suppliers and the computer services companies use the first approach. The independent package suppliers use the second approach.

1. The service-oriented approach
   The service-oriented approach emphasises the service that the supplier provides to the customers, and, not surprisingly, it is the approach taken by those package suppliers whose involvement with applications packages is secondary to their main business activities. Thus, the system suppliers use packages both to help sell their own hardware and to complement their range of software services. If a potential sale is sufficiently important to the supplier, and if the potential customer exerts enough pressure, a system supplier will modify a package to suit the potential customer's requirements in the same way that a computer services company will. Potential purchasers of an applications package can recognise when the supplier is using this service-oriented approach by the supplier's expressed willingness to provide the source code to the purchaser.

   Suppliers use different methods to customise a package to meet the needs of a particular purchaser. Some suppliers will provide their own staff to modify the package, and others will permit either the customer's own data processing staff or the staff of a third party to modify the package.

   The purchaser of a customised package will, however, pay a heavy penalty as regards the subsequent maintenance of the package by the supplier. All the suppliers that we interviewed said they would not support a customised package within the terms of their normal maintenance contract. In addition, the customer has to pay for the modifications necessary to customise the package to his requirements.

   With the exception of Dataskil Limited, all of the system suppliers and the computer services companies that we interviewed use a service-oriented approach. We examined the Dataskil Prosper system, and the representative of Dataskil Limited emphasised that his company does not regard Prosper as an ordinary package. The Prosper system was originally conceived as a language, and it has subsequently been enhanced to become a powerful tool that can be used for building systems. The tool is not modified to meet the requirements of a specific application, but it is used to build the appropriate solution. This simple technical concept permits Dataskil Limited to take a purely product-oriented approach (such as we discuss below) with the Prosper system.

2. The product-oriented approach
   In general, the product-oriented approach is the approach that the independent package suppliers take. These suppliers aim to provide a flexible product backed up by a first-class service both for maintenance and product development. Before they release a new product they increasingly invest resources on research and development to an extent that, at first

sight, would seem to be excessive. They estimate that it costs them between five and fifteen times as much to develop a truly flexible package product as it does to develop an equivalent bespoke system.

There is an important difference between this approach of providing a flexible product and the approach of providing a flexible service. That difference is that once a flexible product enters the marketplace it is not subsequently modified for individual customers. Typically, the suppliers that adopt this product-oriented approach to applications packages do not release the source code of a package to the purchasers of that package.

The rationale of this approach of providing a flexible product is that a supplier cannot successfully support individual customised versions of a package once a fairly small number of customised versions have been created. The livelihood of the independent package suppliers depends on the success of their packages. If they are to survive, they must avoid situations where either they incur unacceptably high maintenance costs or their customers stop using a package because a new release of the package is not sufficiently reliable. Consequently, many independent package suppliers support only the latest release of each package. They achieve this by issuing new releases of a package to all those of their customers that have a maintenance agreement with them. In addition, they insist that a customer who wishes to renew a lapsed maintenance agreement repurchases the latest release of a package.

Both of the independent package suppliers that we interviewed used a product-oriented approach to the supply of applications packages.

THE RANGE OF HARDWARE SUPPORTED BY PACKAGES

We were not greatly surprised to discover that the system suppliers' applications packages are designed to run only on their own ranges of hardware. The computer services companies, on the other hand, usually create an applications package that will run on a popular range of hardware. Once such a package has been established on a popular range of hardware, the supplier may then develop a limited number of versions of the package to run on other ranges of hardware.

The independent package suppliers, however, always assume that their applications packages will be run not only on several different ranges of hardware, but also on different sizes of computer within any particular range. Nevertheless, the independent package suppliers, like the computer services companies, usually have a favoured range of hardware and more often than not that range is IBM's. Consequently, in developing the first version of a package, the independent supplier will normally design it to run on a predefined target machine. For example, the current policy of Management Science America Limited is to design packages primarily to run on the IBM 4300 range of computers in a DOS (or OS) CICS/VSAM environment. In addition, that supplier designs its packages to interface with one or more of the most popular database management systems that exist on that range of computers (in other words, with DL1, Total or IMS).

Applications packages usually run on a wide range of configurations within a particular hardware range. In particular, the input and the output devices that a package uses can be varied according to the particular hardware configuration available. Nevertheless, packages are usually designed for a target machine that is in the middle of a particular hardware range. Although such a package can be run on a minimum hardware configuration (for example, a configuration consisting of less than 64k bytes of memory, one disc, one input device and one output device), it will probably perform inefficiently when it runs on a minimum hardware configuration.

THE RELATIONSHIP SUPPLIERS HAVE WITH HARDWARE MANUFACTURERS

As far as applications packages are concerned, the computer services companies and the independent package suppliers have a dual relationship with the hardware manufacturers (that is, with the system suppliers). On the one hand, they compete directly with the system suppliers, because most system suppliers provide, or intend to provide, applications software. On the other hand, they rely on the system suppliers, because packages are of no value without the hardware to run them on. (Of course, the system suppliers also rely both on the computer services companies and the independent package suppliers, because the availability of successful applications packages may increase the sales of the system suppliers' hardware. That is why the system suppliers have entered into the applications packages marketplace.)

This mutual dependence between the system suppliers and the other two kinds of suppliers of applications packages exists for all types of hardware and all types of packages. In the area of large mainframe computers and expensive (and sophisticated) packages, the relationship is one of mutual courtesy. In that area, the computer services companies and the independent package suppliers normally decide on a well-known target machine for which to design their packages, and they then concentrate on that range of hardware for several years before extending their packages to run on other ranges of hardware. For example, the MAS range of packages was originally designed to run on the ICL 1900 range of computers. The IBM version and the PDP 11 version of the MAS range were developed respectively five years and nine years after the initial ICL version was developed. Similarly, Management Science America Limited (as we have already mentioned) now designs its packages for a standard IBM configuration, and Dataskil Limited designs its products to run only on ICL equipment.

Those of the hardware manufacturers that supply mainframe computers are well aware of the independently-supplied packaged software that is available for use on their own ranges of hardware. For example, IBM now maintains a list of software packages that are available from non-IBM sources and that can run on IBM hardware. Also, a mainframe computer system supplier will sometimes purchase a package developed by an independent supplier, and will then market that package as part of its own proprietary software offerings. Alternatively, a system supplier will sometimes sub-contract both the development and the subsequent maintenance of a package to an independent supplier.

In the area of minicomputers, the relationship between the system suppliers and the other two types of suppliers of applications packages is one in which the system supplier needs the help of the software supplier. The minicomputer manufacturers need this help for two reasons:

- Minicomputer manufacturers are not generally experienced in providing the applications support and the software support that the users of their equipment require.

- Minicomputer sales depend increasingly on the availability of good applications software.

The minicomputer manufacturers' lack of experience in providing applications support and software support has been highlighted by the entry of both IBM and other mainframe manufacturers into the small-computer market. These manufacturers have supported their entry into that market by providing a range of applications packages aimed at the small business user. Consequently, minicomputer manufacturers are anxious to collaborate with those independent software suppliers that already have a proven range of applications packages. For example, Digital Equipment Company Limited has recently made an agreement with Hoskyns Group Limited to market jointly the PDP 11 version of the MAS range of packages. Also, Package Programs Limited has been approached by several minicomputer manufacturers. Each manufacturer has asked Package Programs Limited to make its popular range of accounting packages available to run on the minicomputer manufacturer's range of equipment.

The minicomputer manufacturers find it attractive to collaborate with an independent software supplier, because they themselves can ill afford the costs of applications software research and development. The easiest way for them to ensure that proven applications software is available on their minicomputers is to arrange for independent package suppliers to provide it.

At the present time, the independent software suppliers do not have a relationship with the microcomputer manufacturers because their products do not sell in the microcomputer marketplace. Applications packages of the type that form the subject of this report do not sell in the microcomputer marketplace because they require more complex hardware and software environments than microcomputers can provide. The microcomputer marketplace is, however, well served by low-cost packages, many of which have been developed either by hobbyists or by other enthusiasts.

The price that a microcomputer purchaser is prepared to pay for applications software is an important factor that will, in our view, prevent the independent package suppliers entering the microcomputer marketplace. The purchaser of any applications package will inevitably relate its cost to the cost of the hardware. The cost of a microcomputer is low and so the cost of any applications package will also need to be low. Thus, the supplier of a package will need to sell many copies of it in order to generate an acceptable level of revenue. We do not believe, however, that any independent package supplier will be able to sell sufficient copies of a package, and there are three reasons for this:

- Truly independent microcomputer packages are difficult to construct, because there are so many different makes and models of microcomputers to be catered for. Also, of course, the package supplier would need to maintain many different versions of the package.

- Package suppliers receive a substantial proportion of their revenue in the form of annual licence fees and maintenance contracts. Microcomputer users will want to make only a single (and low) payment for any applications software.

- Software copyright protection is an ill-defined area, and it is difficult to enforce. Unauthorised copying of a microcomputer package could easily prevent the high volume of sales a supplier needs in order to generate the required revenue from a package.

THE APPLICATIONS AREAS SELECTED BY THE SUPPLIERS OF PACKAGES

Packages are available for a wide variety of applications areas, but most suppliers of packages concentrate on just a few selected applications areas. The reason for this is that a package supplier has to develop expertise in the selected areas, and so only the larger mainframe system suppliers can afford to develop packages in many different applications areas. For example, IBM and ICL provide packages in most applications areas, including some highly-specialised and rarely-used applications. For highly-specialised applications areas, IBM often purchases a successful system from one of its customers. (IBM's agreement to market Lockheed's graphic design system is an example of this approach.)

On the other hand, and for the reason just given, the computer services companies and the independent package suppliers prefer to develop expertise in a limited number of applications areas, and they are unwilling to develop packages for other applications areas. Instead, they prefer to develop new versions of their existing packages to run on different ranges of hardware.

Our interviews with package suppliers indicated that in selecting an applications area for which to develop a package they apply three criteria:

- The applications area should be sufficiently general to provide an adequate market for a package.

- The applications area should be sufficiently complex to prevent others developing cheap and quick solutions that will satisfy the needs of the majority of potential end users. The applications should also be sufficiently complex to make the cost of developing a bespoke system beyond the means of all but the largest organisations.

- The applications area should be sufficiently well defined by either legislation (for example, taxation) or professional and international standardisation (for example, accountancy or production control).

The majority of applications packages available in the marketplace are in a few well-defined applications areas. Our analysis of the United Kingdom marketplace in chapter 2 showed that accounting and financial applications packages are provided by many applications packages suppliers. Manufacturing and production control applications packages are also available from several suppliers. But the applications areas in which the widest choice of packages exist are payroll applications, pensions applications and the personnel applications associated with both payroll and pensions. Any package supplier that is a serious contender in the applications packages marketplace inevitably has its own payroll system to offer to potential purchasers.

THE ORIGINS AND THE DEVELOPMENT OF PACKAGES

All of the package suppliers that we interviewed have their own systems development staff, although not all of the products they provide necessarily originate from their own workshops. Several packages are sold and supported by agencies, because only the largest package developers can afford the international marketing costs that are necessary to create a worldwide market for a package. For example, before an international Q-PAC company was formed in the late 1970s, the Q-PAC payroll package had been provided exclusively through several international agents for more than ten years. Also, several of the packages provided both by Package Programs Limited and RTZ Computer Services Limited originated in the United States, and are now marketed in the United Kingdom under licensing agreements.

The mainframe system suppliers have their own software development sections which develop most of their applications packages. As we mentioned on page 18, however, they are prepared to buy in packages where this is appropriate. On the other hand, Digital Equipment Company Limited openly admits that it does not develop packages. Instead of developing packages, Digital Equipment Company Limited prefers to negotiate joint agreements with software companies, and the agreement it has with Hoskyns Group Limited for the MAS range of packages is an example of this approach.

In general, the computer services companies and the independent package suppliers develop their own applications packages. Sometimes, however, they are prepared to market a package developed by another supplier, provided that it complements their own range of products. There have even been instances where an independent supplier has taken over another package supplier. For example, when Management Science America Inc. was looking for a suitable international payroll package to market, it decided to acquire the whole Q-PAC company instead of merely acquiring the marketing rights to the Q-PAC package.

Package suppliers usually develop a series of interlocking packages within a particular applications area, and the suppliers and their customers alike benefit from this approach. The suppliers benefit because the series of interlocking packages provides them with multiple and on-going sales opportunities. Their customers benefit because they can gradually introduce packages into the particular applications area. Both the MAS range of packages and the products

marketed by Management Science America Limited are designed as interlocking ranges of packages. Sometimes, a package supplier develops a series of interlocking packages that spans several different, but related, applications areas. Thus, a package in one applications area can sometimes be interfaced to a package in another area, provided that both packages are supplied by the same supplier.

Apart from the interfaces that are provided between different packages in an interlocking range of packages, applications packages are not, in general, designed from the outset to interface with other applications systems. Nevertheless, the majority of package suppliers now develop their products to interface with the most widely used database management systems. Also, links between such packages and other applications systems can be provided by using the database management system as an intermediary. Even when a database management system cannot be used in this way, a package can easily be linked to other applications systems by constructing an interface consisting of a bespoke file and program.

Even though a package is normally marketed for use in a particular applications area, it may well have been developed so that it can be used also in several other applications areas. For example, a well-designed general ledger package could easily be modified to enable it to be used in any application that requires complex hierarchical data structures.

A successful applications package will be used by many diverse end users, and those responsible for designing and developing a package need to ensure that it is as flexible as possible. The earliest method used to develop a flexible package was known as the "preprocessor-customiser" method. With this method, the end user selected the parameters required to customise the package, and the parameters and a general version of the package were then processed by a customiser program to generate the customised version of the package. IBM and other package suppliers formerly used the preprocessor-customiser method, but, today, flexible packages are mainly developed by using two alternative methods. These two methods, which we discuss below, are the parameterisation method and the open-ended design method.

1. The parameterisation method
   With the parameterisation method, the complete range of options that the package provides are built into the package, and the end user uses parameters to select his required options. Options are normally available both for processing functions and files, and the files the end user requires are specified by the appropriate parameters. The parameters that specify the required processing functions are processed as the package is run, and so those functions that the end user does not require are bypassed. Most of the commercial packages available today use the parameterisation method to provide the required flexibility. Parameterised packages are usually designed in such a way that the applications system can easily be modified to satisfy changed end-user requirements by specifying a new set of parameters.

2. The open-ended design method
   The open-ended design method of developing packages provides a basic applications processing framework that permits additional user-specified routines to be appended to the basic package. The method has one main disadvantage. To use an open-ended package to implement an applications system, an organisation requires specialist data processing support to develop the additional routines. The method has the advantage, however, that the resultant applications system is likely to be nearer to a bespoke system than is a system developed with the use of a parameterised package.

In addition to the parameterised method and the open-ended design method, a third method is sometimes used to develop a flexible applications package, and this third method is the one the Prosper system employs. (The Prosper system is a sophisticated flexible system building tool that permits the user to create his own tailor-made system by using the standard functions and the standard file-handling mechanisms that the system provides.)

All of the package suppliers that we interviewed recognised that the time, the cost and the effort required to develop a package are all several times higher than those required to develop an equivalent bespoke system. There is, however, a distinct difference between the development approach of the service-oriented suppliers and the development approach of the product-oriented suppliers. The service-oriented suppliers approach the development of a package in a way that is similar to the approach used for developing bespoke software. The product-oriented companies, however, recognise that, for two reasons, a true package is fundamentally different from bespoke software. The first is that a true package needs to be more flexible than is bespoke software, and the second is that a true package should be developed from the outset to run both in different end-user environments and in different hardware and software environments.

To develop packages that are flexible and that also can run in different environments, a package supplier needs to take a different approach to the one used for developing bespoke software. The "blueprint" development approach that Management Science America Limited now uses for developing packages illustrates the approach that, in our view, product-oriented suppliers need to take. With the blueprint approach, the development of a package commences with a research stage. In that stage, the functions that the package is to provide are determined, as are the basic technical features that the package will use. In addition, the target hardware and software environment is specified for the first version of the package. At the end of the research stage, the first version of the package is then developed, and this first version is then used as the blueprint version to generate alternative versions of the package for different hardware and software environments. The mechanism that is used to develop alternative versions of a package is a filter that is oriented to a target machine. The filter translates the blueprint version of the package into source code that can be compiled and then executed on the target machine.

The blueprint approach requires the package developer to invest substantial amounts of both time and money in the development of a package. Those package suppliers that either do not have, or are not prepared to make available the resources to make the investment that the blueprint approach requires, tend to redevelop a package at least once, and perhaps twice. The first redevelopment usually takes place about two to three years after the initial version of a package has been developed. Sometimes, in redeveloping the package, the supplier is able to enlarge the scope of the package as a result of the experience gained from the initial version. At a later stage, a supplier will sometimes redevelop a package a second time to enable the package either to exploit new technical features or to run in a completely different software and hardware environment.

SUMMARY

In this chapter, we have shown that the suppliers of applications packages can be divided into two main groups. On the one hand, there are those suppliers that take a service-oriented approach to the supply of applications packages. In general, those suppliers consider that by providing applications packages they are providing a service which either complements or adds to other services or products that they supply. On the other hand, there are those suppliers that take a product-oriented approach to the supply of applications packages. In general, the commercial success of those suppliers depends primarily, or even exclusively, on the revenue they derive from supplying applications packages.

It is well known that the cost of software, relative to the cost of hardware, forms an increasingly larger proportion of the total costs of information processing systems. This trend means that the revenue from software that the system suppliers and the computer services companies receive will increasingly form a larger proportion of their total revenues. For this reason, and for the three other reasons we give below, we believe that the system suppliers' approach and the computer services companies' approach to applications packages will increasingly shift from being service-oriented to being product-oriented. The three other reasons for this shift are:

- Applications packages will increasingly become more attractive than bespoke systems, so far as end users are concerned. The reason for this is that, as the cost of developing bespoke systems continues to increase, the cheaper solution that applications packages offer will become irresistible.

- Applications packages will provide a greater proportion of the profits both of the system suppliers and the computer services companies. This increase in the proportion of profits will come partly from an increased volume of sales of packaged software, and partly from a relative fall in hardware profits.

- Higher volumes of sales of applications packages will substantially increase the maintenance burden that service-oriented suppliers have to bear. The service-oriented suppliers, therefore, will be forced both by commercial pressures and management pressures to move towards a product-oriented approach to supplying applications packages.

We believe, therefore, that the applications packages suppliers will, in future, become predominantly product-oriented suppliers.

When the majority of suppliers have become product-oriented, this changed approach will have several effects both on the suppliers themselves and on their products. In our view, the most important effects will be:

- There will be a smaller number of applications packages suppliers, because fewer organisations will be prepared to make the substantial investment that is required to research, develop and market each new product.

- The existing package suppliers will extend their range of products slowly, if at all, because of the substantial investment that is required to launch a new product.

- A greater proportion of packages will be standard products, which their suppliers will not be prepared to modify for individual users.

- User groups will play an increasingly important role in deciding which new features will be incorporated in new releases of a package.

- The end users will continue to be the suppliers' major sales target, and this will mean that suppliers will provide excellent end-user documentation and operating documentation.

- Suppliers will not supply their users with the source code of a package. The result will be that, increasingly, maintenance will be carried out by the supplier, rather than by the user.

- Suppliers will build greater flexibility into packages, so that end users can configure the package to meet their particular needs.

- An increasing proportion of packages will be developed to run in a standard hardware and software environment. A supplier then will not provide versions of a package to run in other environments unless, in the supplier's view, the potential market size justifies the effort involved in creating a new version.

In conclusion, the product-oriented suppliers of applications packages usually have a limited number of high-quality products, and they provide excellent all-round maintenance and

support. In our view, these product-oriented suppliers provide both a better product and a better service to their customers than do the service-oriented suppliers of applications packages.

CHAPTER 4

FOUNDATION MEMBERS' EXPERIENCE WITH APPLICATIONS PACKAGES

As part of our research for this report, we investigated both the experience that Foundation members have of using applications packages and the opinions they have about the marketplace for applications packages. In particular, we undertook a detailed survey of the experience of those members that are based in the United Kingdom. The survey took the form of a questionnaire which asked members to provide information on the following topics:

- The current use members make of applications packages.
- The patterns of members' expenditure on systems development.
- The criteria members use in selecting applications packages.

Thirty-two Foundation members returned wholly completed (or partially completed) questionnaires, and the respondent organisations represented a wide range of industry and business sectors. In addition to conducting the questionnaire survey, we interviewed a selection of the respondent organisations, and we also conducted interviews with several organisations (both from Continental Europe and the United States) that use applications packages.

The findings of our research into the use of applications packages are reported in this chapter under three main headings:

- The role of applications packages in systems development.
- The evaluation and the installation of applications packages.
- An analysis of the packages used.

THE ROLE OF APPLICATIONS PACKAGES IN SYSTEMS DEVELOPMENT

We examined four aspects of the role of applications packages in the systems development process, and we now set out the findings of this examination. Firstly, we review the expenditure on applications packages compared with the expenditure on other approaches to systems development. Secondly, we review the priorities that various organisations give to applications packages in relation to several other ways of improving the systems development process. Thirdly, we report on the impact that applications packages have on the choice of computer equipment, and fourthly, we conclude with the main comments members made concerning the use of applications packages in large and/or mature data processing installations.

Expenditure on systems development approaches

Members were asked to indicate their approximate levels of expenditure on the following five different approaches to providing applications systems:

- In-house systems analysts and programmers, including their direct overheads.

- Software houses providing analyst and programmer services.
- Turnkey systems.
- Applications packages, including licence fees and maintenance charges.
- Applications packages run on external computer bureaux.

Figure 8 summarises the expenditure patterns that emerged from the returned questionnaires. Not surprisingly, the overall pattern shows that organisations spend most on developing and maintaining bespoke systems, using either in-house or contract staff. Two-thirds of the respondents spend at least ten times as much on developing and maintaining bespoke systems as they do on applications packages, and four-fifths spend less than £50,000 per annum on applications packages.

One surprising finding from this part of the survey was that only 17 per cent of the respondent organisations spent more than £50,000 per annum on turnkey systems (£50,000 being the

Figure 8 Patterns of expenditure on systems development approaches

Class of expenditure	Percentage of respondent organisations whose expenditure (in £000s) falls into each of the ranges shown below					
	0 to 10	11 to 50	51 to 200	201 to 500	More than 500	Information not provided
Expenditure on in-house systems analysts and programmers, including their direct overheads	0	0	9	25	59	7
Expenditure on software houses providing systems analysis and programming services	31	22	28	6	6	7
Expenditure on turnkey systems	68	9	9	4	4	6
Expenditure on applications packages, including licence fees and maintenance charges	28	50	13	2	2	5
Expenditure on applications packages run on external bureaux	50	13	19	13	0	5

typical cost of a small turnkey system). During the interviews, however, several organisations mentioned that, in the past few years, end-user departments have increasingly purchased turnkey systems direct from the supplier. As a consequence, the amount spent on turnkey systems by the information systems department is not always an accurate guide to the total amount spent by an organisation on turnkey systems.

Ways of improving the productivity of systems

Member organisations were asked to rank, in order of importance to themselves, a number of possible ways of improving the productivity of systems. The purpose of this question was to assess the importance that members place on applications packages in comparison with other ways of improving the productivity of their systems.

We analysed the replies to this question in two ways, and the results are shown in figure 9. The first column in the figure shows the number of respondent organisations that ranked a particular way as their top priority for improving the productivity of their systems. The second column shows the percentage of respondent organisations that ranked a particular way as one of their top three priorities. From the results of these analyses, we conclude that the availability of more and better applications packages would be regarded as a welcome improvement by most member organisations, but that only a few members see the use of applications packages as being their highest priority for improving the productivity of their systems. Foundation Report No. 25, which will be concerned both with approaches to and methods and techniques for developing systems, will examine the area that most members perceive as the most important area for improving the productivity of systems.

Figure 9 Applications packages and the productivity of systems

Way of improving systems productivity	Number of respondents who ranked a particular way as their top priority for improving the productivity of their systems	Percentage of respondents who ranked a particular way in their top three priorities for improving the productivity of their systems
Improved design and development methods	9	75
More and better tools to enable the user to help himself	5	53
More and better applications packages	4	53
Better trained analysts and programmers	4	47

Applications packages and computer selection

The questionnaire asked members to comment on how importantly they regard the availability of applications packages when they select either computer hardware or a bureau service. Twelve organisations had selected a computer either because of the availability of a specific package or packages, or because they knew that a wide selection of applications packages was available for use on that range of computers. The respondent organisations mentioned 17 different types of computers (mainly small business systems, minicomputers and personal computers), but there was no clear indication that different organisations had selected a particular type of computer because a specific package was available on that computer. No one type of computer was mentioned by more than three respondents, and only two types were mentioned more than once.

A majority of respondent organisations (about 66 per cent) use bureaux services because they offer particular applications packages. Several respondent organisations said that they used a bureau service for the following three types of applications:

- Engineering, scientific and mathematical applications (mentioned by 11 respondent organisations).
- Financial planning applications (mentioned by five respondent organisations).
- Payroll applications (mentioned by four respondent organisations).

One-third of the respondent organisations spent more on running packages on external bureaux than they spent on licences for packages that they ran on their own computers. We suspect, however, that respondent organisations use bureau facilities to a greater extent than they reported to us. We say this because we found, during our interviews with users of packages, several examples of line departments that use bureau facilities without reference to the information systems function.

The use of applications packages in large and/or mature installations

Several members expressed the view that packages were most appropriate for, and were most readily accepted by, first-time end users in their organisation. Those end users who have previous experience of using computer-based systems frequently regard their requirements as being too complex to be processed by a packaged solution.

Members who expressed this view believed that packages were more appropriate for new installations that were at the stage that Nolan has termed Stage 1 of the evolutionary cycle of using computer systems. (A description of Nolan's Stage Hypothesis can be found in Foundation Report No. 11 on pages 6 and 7.) Those same members believed that few package suppliers provide suitable products for an organisation that has passed beyond that stage. In particular, they considered that packages were of little value to those organisations that have either some degree of integration between applications systems or a data management strategy that requires a database approach.

If packages are to be of value to more mature data processing organisations they need to have a new form of inbuilt flexibility. For example, some respondent organisations would like to use packages in situations where they will need to interface both with other applications systems and with environmental software (such as database management systems and teleprocessing monitors). Packages, therefore, need not only to have the functional flexibility to make them suitable for several organisations to use (which is a prerequisite for the commercial success of the package suppliers). They need also to provide flexible links both to other applications and to environmental software. For example, it should be possible for an organisation to remove the

file management subsystem from a package and replace it with a subsystem that enables the package to interface with a database management system.

Two members that we interviewed believed that their data processing strategy precluded the need for packages. Their strategy is to:

- Manage data as a resource, using data dictionary and database management software.
- Provide transaction processing software for the end user.
- Train end users in the use of an enquiry language (such as Filetab or Easytrieve), so that the end user himself can provide most or all of the output routines.

Those two members have both based their strategy on the same rationale. They believe that transaction processing and file management should be provided by a professional systems development team because:

- Systems concepts and performance constraints need to be understood if these tasks are to be performed properly.
- Requirements for transaction processing and file management with most applications are very stable, and so they involve comparatively little maintenance work.

Conversely, they believe that many of the requirements for information from a system will change from time to time (for example, when a new manager is appointed in an end-user department), and the system will be required to provide new reports for the end users. The preparation of new reports does not, however, require an understanding of systems concepts. Moreover, the penalties (in terms of computer performance) are unlikely to be severe if the reports are not produced in the most efficient manner. The two members believe, therefore, that much of the "back-end" of the system can be left to suitably trained end-user staff to provide for themselves.

THE EVALUATION AND THE INSTALLATION OF APPLICATIONS PACKAGES

In the questionnaire survey, Foundation members were asked several questions concerning the evaluation and the installation of applications packages.

Twenty-nine of the respondent organisations said that they normally evaluate packages as a matter of course during the feasibility stage of a development project. Many members indicated that they had adopted this policy only a short time before. Only five members, however, reported that they had a formal set of selection criteria, although several members listed the factors that they regard as being particularly important in the selection process. The three factors they quoted most frequently were:

- The existence of an established customer base to which reference could be made (quoted by 33 per cent of the respondent organisations).
- The availability of the source code for the package (quoted by 22 per cent of the respondent organisations).
- A stable supplier that has sufficient resources and adequate capability to support the package (quoted by 20 per cent of the respondent organisations).

Members were asked to give their principal reasons for selecting an applications package. The

two most commonly quoted reasons, each quoted 23 times (that is, by more than 70 per cent of the respondent organisations), although not always together, were the greater speed with which a system could be implemented and the reduced cost of development compared with a bespoke system.

Other reasons for selecting an applications package that several members quoted were:

- The guarantee that modifications will be carried out by the supplier to cater both for statutory changes and for new requirements identified by several users (quoted by 12 per cent of the respondent organisations).

- The delegation of maintenance work to the supplier (quoted by 12 per cent of the respondent organisations).

- A reduced risk that a development project will deviate from the planned cost or time, because the system uses tried and tested software (quoted by 12 per cent of the respondent organisations).

- The ability to provide a flexible system able to meet unforeseen business needs in the future (quoted by 10 per cent of the respondent organisations).

Members were also asked to give their views on who should be responsible for modifying and maintaining applications packages. Two contrasting views were expressed, with two-thirds of the respondent organisations being willing to modify packages themselves (or to have them modified on their behalf), and one-third of the respondent organisations refusing to make changes to packages at all costs. About half of the respondent organisations try to avoid modifying a package until at least a minimum period of time has elapsed, and some respondent organisations saw the use of unmodified packages as a convenient method for prototyping certain types of system. About 55 per cent of the respondent organisations rely on the supplier to maintain the package, although many of those organisations safeguard their position by familiarising their own staff with the details of the system.

During the interviews, many organisations told us that they believe they are now more likely to find suitable applications packages than they would have been able to a few years ago. This changed expectation has led them to survey the package market as a matter of course during the feasibility stage of a project.

The marketplace for applications packages is immature, however, and the claims of few, if any, suppliers can be accepted at face value. Even so, no package user that we interviewed felt that he had a sound methodology for evaluating packages. Several Foundation members, however, were particularly concerned about the problem of trying to assess, in advance, the hardware facilities that a package will eventually require in operation. They said that they had found that a package had made greater demands for computer time and computer resources than they had expected when they selected the package.

Several members also commented on the role of the end user in the package evaluation process. They quoted examples where the end user (usually an accountant) had undertaken the evaluation of a package without reference to the information services function. Often, as a result, the end user had selected a package that was either functionally inadequate or operationally inefficient.

An increasing number of organisations are implementing some form of distributed processing, and a distributed processing environment creates a potential new demand for packages. At the moment, the most common form of distributed processing exists in those organisations that have installed either identical or similar systems at several locations. With this approach the

organisation needs either to purchase a suitable software package, or to develop an in-house package that the several locations can all use.

Many organisations have set about developing an in-house package by using one typical location for the first implementation of the package. They then modify the package later to meet the requirements of the several other locations. Foundation members that have used this approach to develop an in-house package have found two drawbacks. First, several organisations have found that an applications package purchased from an outside source is much more likely to be accepted by end users than a system designed for another part of the organisation. In particular, if many other organisations are already using the package, the end users in an organisation that are selecting that package will tolerate any shortcomings they identify in the facilities provided by the packaged software. By contrast, those same end users will not tolerate similar shortcomings in an in-house package.

Second, the strategy of developing a system for one part of the organisation, and later modifying it for other parts, places the systems development department in precisely the same position as a software house that launches a package based on a generalised version of a bespoke system. In their enthusiasm to establish their new package with several users, the software house and the systems development department in an organisation find that they soon have separate versions of the software for each user. They find too that these separate versions create an unacceptable maintenance load. The software house, of course, has a commercial incentive to redevelop the software as a proper package. But the systems development department in an organisation has no incentive to redevelop the software because often the department's performance is measured by the amount of additional systems capability it provides to end users. No matter how appropriately and how effectively the department redevelops the software, the redeveloped package will not provide any additional systems capability.

In theory, organisations could decide to develop a proper package in the first place. However, before embarking on such a course of action, management needs to be aware of the following facts:

- As we discussed in chapter 3 on page 16, the development of a proper package costs between five and fifteen times as much, and takes at least twice as much time, as the development of a single bespoke system.

- The design of packaged software requires the use of very scarce package design skills and experience.

Several users of packages whom we interviewed made comments of general concern relating to the installation of applications packages, and we summarise below their four main comments:

1. Project management
   Without a proper project management methodology, projects that use packages are as likely to experience delays and cost overruns as are projects that use bespoke software.

2. Test data
   Unless the supplier provides a set of test data, the task of creating test data (and producing the subsequent results) is a substantial one, and, typically, it occupies a fairly long elapsed time. This means that it will probably be on the critical path of the development project.

3. Job control data
   The preparation of operational job control data can be a major project in itself. This is particularly so if the job control data is used to provide the security that the installation normally builds into its applications programs.

4. Package modifications
Packaged software should be modified only as a last resort. The preferred methods for handling possible modifications to a package are set out below, with the most frequently mentioned methods listed first:

- Avoid modifications at all costs.
- Have the package supplier incorporate the modifications and subsequently maintain them (if he is prepared to do so).
- Put pressure on the supplier (perhaps through the user association) to incorporate the required modifications in a future release of the package.
- Provide the additional facilities by building special-purpose software that interfaces with the package. This commonly-taken approach does, however, have drawbacks. We were told of several cases in which changes to the package itself, although they were transparent to the end user, involved modifying the purpose-built software.

Members were also asked to provide details of the applications areas for which they had considered and then rejected the use of packages. The pattern that emerged from our analysis of the returned questionnaires is summarised in figure 10. The most frequent reason (which two-thirds of the respondent organisations quoted) for rejecting a packaged approach was that the packages they evaluated failed to meet their business requirements. Several respondent organisations also gave the following four other reasons for rejecting a packaged approach:

- The problem of interfacing a package with other applications systems and/or with environmental software (quoted by 25 per cent of the respondent organisations).
- The limited choice of packages available for a particular range of computers (quoted by

Figure 10 Applications areas for which potential users most commonly reject the use of packages

Applications area	Percentage of respondent organisations that have rejected the use of packages	Percentage of respondent organisations that use packages
Order entry and invoicing	32	23
Job costing	29	23
Pensions records	29	19
Personnel records	26	19
Employee benefits	23	3

Figure 11 Applications for which organisations commonly use packages

Applications area	Number of respondent organisations that reported the use of packages in the business sectors shown below									Percentage of all respondent organisations
	Food processing	Electronics	Energy/primary	Engineering	Beverages	Retailing	Chemicals	Other (1)	All sectors	
Financial planning	6	4	4	4	1	1	2	3	25	78
Payroll	3	5	4	3	2	1	1	1	20	63
Nominal ledger	6	4	1	4	1	–	1	2	19	60
Sales ledger	6	5	1	3	–	–	–	3	18	56
Assets accounting	4	5	3	1	–	1	–	1	15	47
Purchase ledger	4	5	1	2	1	–	–	2	15	47
Bills of materials	3	5	2	3	–	–	–	–	13	41
Project management	2	1	2	2	1	2	–	2	12	38
Inventory control	2	5	–	2	–	1	–	–	10	31
Production planning and control	1	3	3	3	–	–	–	–	10	31
Telephone usage	4	2	1	–	1	1	–	1	10	31
Statistics	2	–	3	–	–	–	2	1	8	25
Computer-aided design	–	2	–	1	–	1	1	2	7	22
Job costing	2	4	–	1	–	–	–	–	7	22
Order entry	3	3	–	–	–	1	–	–	7	22
Pensions records	1	2	1	–	1	1	–	–	6	19
Personnel records	1	3	1	1	–	–	–	–	6	19
All other applications (2)	4	10	1	–	–	1	–	2	18	56
Number of respondent organisations	6	5	4	4	3	3	2	5	32	
Average number of packages per respondent organisation	9	14	7	8	3	4	3	4	7	

Notes:

1. This sector includes two government departments (although information provided by one of them, the CCTA, represents more than 300 computer installations and is not included in this table), and one respondent organisation from each of the construction, the financial and the travel sectors.
2. For all other applications areas, packages were used by fewer than five respondent organisations. Purchasing packages, however, were used by four respondent organisations from the electronics sector.

12 per cent of the respondent organisations, all of whom had either ICL or Burroughs equipment).

- The comparatively low level of support available from package suppliers (quoted by 10 per cent of the respondent organisations).

- The high cost of the package they evaluated, caused, in their view, by unnecessary flexibility and/or sophistication in the package (quoted by 10 per cent of the respondent organisations).

ANALYSIS OF THE PACKAGES USED

As part of our survey, Foundation members were asked to provide details of the applications packages their organisations currently use, or are likely to use. Figure 11 provides a summary of the use of packages they reported, and it shows the number of respondent organisations in each business sector that use packages in each of the applications areas. Not surprisingly, financial applications packages are the most commonly used. (More than half of the respondent organisations, in fact, use applications packages only in the financial area.) Packages relating to production planning and control (including packages for bills of materials and inventory control) were used by only about one-third of the respondent organisations, but, of course, not every business sector requires this type of application.

The results of the analysis, as shown in figure 11, need to be treated with some caution, because they relate to only a small sample of users. On the basis of this analysis, however, each respondent organisation used, on average, seven applications packages. The electronics industry (including several suppliers of computers or computer-related products) makes the most use of applications packages (an average of 14 packages per respondent organisation in this industry), and the food processing and the engineering industries both make higher-than-average use of packages.

Members were asked to provide us with the names of the packages their organisations use, or plan to use. About 80 per cent of the respondent organisations provided this information, and the most frequently mentioned packages are listed in figure 12 overleaf. As the figure shows, only four packages were mentioned by more than five different respondents.

SUMMARY

Most Foundation members are now much more prepared to consider a packaged solution than they were in the past, and they now make use of applications packages wherever possible. The most common reason that members gave for using an applications package was to reduce the time taken to implement an applications system, although, in some organisations, the lack of systems development staff meant that a package was the only way that a particular system could be implemented in the required time.

Whilst applications packages can reduce the cost of implementing a system, several of the organisations that we interviewed said that they had seriously underestimated the effort required to install a package. Those organisations said that they had learnt that the process of installing a package needs to be managed in the same way as any other part of the systems development process.

Figure 12 The packages used most frequently by respondent organisations

Applications area	Number of respondents who reported the use of a package	Package name and the number of respondent organisations using that package	
		Name	Number of users
Financial planning	25	PROSPER	8
		EPS/FCS	7
Payroll	20	UNIPAY	10
		Q-PAC	5
		K-PAYE	4
Nominal ledger and general ledger	19	MMS	3
		MSA	3
Sales ledger	18	NOR-SAL	4
Assets accounting	15	FASCIA	6
Bills of materials	13	MAPICS	3
Project management	12	PERT	3
Production planning and control	10	OMAC	3
Telephone usage	10	TIGER	4
Pensions records	6	UNIPENSION	3

Note: No other package was mentioned by more than two respondents.

CHAPTER 5

SELECTING AND INSTALLING APPLICATIONS PACKAGES

In chapter 2, we indicated that the use of packaged applications software was accepted in the United States earlier than it was elsewhere. This earlier acceptance was most probably explained by two factors:

- Computer technology has been applied in the United States at a faster rate than it has in Europe, and the subsequent demand for computer applications has outstripped the supply of analysts and programmers to an even greater extent than it has in Europe.

- The United States marketplace is very much larger than the fragmented markets in Europe taken together.

As a consequence, suppliers that have entered the applications packages market in the United States have had better prospects of achieving commercial success than have their counterparts in Europe. Package suppliers in the United States have therefore had a greater incentive to invest in products that are designed and constructed from the beginning as packages. This approach contrasts with the approach frequently met in Europe, where applications software that was conceived originally as a once-off implementation is later generalised to create a package. In addition, user organisations in the United States have, in general, become more accustomed to evaluating and installing packages than have user organisations in Europe, and this greater willingness to consider packages has encouraged the development of the applications packages marketplace.

Even so, the European market for packages is now more mature than it was, and during the research for this report, many Foundation members indicated to us that they are now more prepared to consider using packages than they were two years or so ago. Most Foundation members now consider the use of applications packages at the feasibility stage of a project, although very few members have any formal policies, or procedures or standards regarding packages. At the same time, there is little evidence that organisations are arranging for one or more individuals to acquire either a familiarity with the packages market or the skills required to evaluate and purchase packages.

We share the view of the many Foundation members who believe that organisations will make greater use of applications packages in the future than they have in the past. For this to happen, however, organisations need to ensure that the tasks associated with selecting and evaluating packages are undertaken as effectively as possible. In this chapter we identify those tasks, and we present them in relation to the development project management structure and, in particular, to the task of defining the end users' requirements. We then present a series of checklists that an organisation can use when evaluating a package, and we also consider the roles both of end users and systems staff in the evaluation process. Finally, we conclude the chapter with a section in which we review the contractual points that an organisation needs to consider when purchasing a package.

APPLICATIONS PACKAGES AND PROJECT MANAGEMENT

The need to use an effective project management structure when developing computer-based

systems is now well understood, as are also the principles on which that structure should be based. The two most important principles are:

- To set clear objectives that the system should meet, and to define clearly both the scope of the project and the responsibilities of those who will develop the system.

- To subdivide the total project into several stages, so that the end of each stage is clearly defined by at least one tangible item (or result) that can be inspected for quality.

Based both on these principles and on the traditional (and essentially analytical) approach to systems development, a standard project management framework has evolved. The five main stages of this standard framework are:

1. The feasibility study stage
   The feasibility study stage is the stage at which a preliminary investigation is made to identify both the nature and the scope of the problem or the opportunity, and also to assess the scale of the assistance that is required, and the benefits that are likely to accrue as a result of developing the system.

2. The definition of requirements stage
   The end users' requirements are identified in detail at this stage, and the objectives of the new system are defined.

3. The construction stage
   At this stage, the system is brought into physical being. This requires the preparation both of programs and clerical procedures, and of the various levels of tests that are required to prove that the system can be expected to meet the objectives set.

4. The implementation stage
   During this stage, the system is commissioned and becomes a fully productive system.

5. The maintenance stage
   In this final stage, the system is maintained for the duration of its operational life. During the whole of that time, any errors found are eliminated, and any additional or changed requirements are catered for by the provision of new or enhanced functions.

In most of the systems development projects that we reviewed during the research for this report, we found that the applications packages had been evaluated within this framework or some variation of it. This meant that packages were normally evaluated after the detailed definition of requirements stage. When this approach is followed the case for using an applications package can be severely reduced, because it involves a fruitless search for a package that meets all of the detailed requirements (including those features that are, to a greater or a lesser extent, desirable but not essential).

A minority of the systems development projects that we reviewed, however, used project management approaches that were different from the traditional analytical approach, and most of those different approaches were similar to a prototyping methodology. Two principal reasons were quoted for using a prototyping approach for systems development projects that made use of applications packages, and they were:

- The realisation that the use of a package was the only way in which a system would be provided. For the projects we reviewed, the analytical approach was not used, mainly because of either a shortage of systems development staff or a tight time schedule for developing the system.

— The recognition by the end user first, that he did not know precisely what his requirements were and second, that a package might provide a satisfactory interim solution. An example quoted to us concerned a fixed assets accounting application where a group of accountants were planning to apply current cost accounting principles of which they had no previous experience. They accepted that a well-established package would probably have most, if not all, of the important features that they might require. Also, they realised that, at worst, a package would enable them over a period of one or two years to define their requirements for a replacement system. At best, by contrast, it would enable them to avoid developing an in-house system that subsequently might prove to have been given insufficient consideration.

DEFINING THE SYSTEM REQUIREMENTS

Whichever of the two systems development approaches is adopted, the system requirements will need to be defined before the available packages are evaluated. For the prototyping approach, the definition of requirements stage will be less rigorous than for the analytical approach. In this section, though, we concentrate on the analytical approach, and we describe the method we recommend for defining the requirements in a way that permits potential packages to be evaluated in an unrestricted way. This method can be modified easily to suit other circumstances.

An important aim of the definition of requirements stage is to establish which of the following development strategies is most appropriate:

— A system constructed completely from purpose-designed software.

— A system based exclusively on packaged software.

— A system constructed from a combination of packaged software and additional, or replacement, purpose-designed software.

Many systems analysts, and end users too, eliminate the second and the third of these development strategies because of the way in which the requirements are defined. Research in the United States indicates that the organisations that make the most use of packages are those that formally distinguish, at the definition of requirements stage, between the essential features and the desirable features required in a system.

One way of making this distinction is to establish a list of the basic requirements that any solution must provide, and this list would include such items as:

— The minimum level of improvement to be obtained from the introduction of the system.

— The things that must, or must not, be changed as a result of introducing the system.

— The regulations, standards and policies that a solution must comply with.

— The maximum life-cycle costs of the system. (Life-cycle costs take account of the costs of development, operation and maintenance.)

Such a list can contain the commercial requirements and also those system requirements that are of paramount importance.

In our view, it is important that an organisation defines the system requirements in a way that does not automatically rule out the use of applications packages. We say this because we

believe that there are now several areas in which applications packages provide a better alternative than purpose-designed software, and we discuss two of them below:

1. The greater flexibility of packages
   Although purpose-built software caters for some of the overall requirements of an applications area, it is designed to meet the needs of a specific customer. Purpose-built software is, in general, inflexible in terms of its file structure, its programs and its interfaces with the end user. This inflexibility largely arises because the overall (or general) requirements of the applications area are treated on a level equal to the particular requirements of the specific customer. On the other hand, applications packages provide a full system, and they are designed to cater for the overall requirements of an applications area, with variations to accepted practices being catered for by the use of built-in options. This approach provides a much higher level of flexibility than is usually provided by purpose-designed software.

2. The better quality end-user interfaces of packages
   Successful packages more often than not provide interfaces that are oriented towards the requirements of end users. These interfaces are concise and complete, and they are specified in terms that the end user is familiar with. The end user is, therefore, able to use the interfaces to control the system and its options. In contrast, purpose-designed software often has less-than-adequate interfaces with the end users. These poor interfaces create a communications gap that prevents the end users from using the full range of facilities available within purpose-designed software. Purpose-designed software seldom provides a high-level interface that permits the end user to reconfigure his system. At a lower level, the interfaces are often incomplete, inadequately specified, ambiguously defined and stated in terms unfamiliar to the end users. As a consequence, the end user requires the constant support of professional data processing staff.

   Because a good applications package uses the end user's terminology, it enables him to understand clearly the options that the package provides, and also the limitations of the controls and the facilities that are available for his use. Many end users are often prepared to forgo their unique individual requirements for the benefit of knowing that, within the limits of the options provided by an applications package, they can control their own systems.

The definition of requirements will normally include descriptions both of the functional requirements and the data requirements of the system. The level of detail in which these requirements are described depends, in part, on the likelihood that a package will be the chosen method of providing the system. If an organisation is likely to use a package, then it is best if the organisation undertakes at this stage some of the system design activities that would normally take place during the construction stage.

For example, the development of a conceptual data model is now becoming an established part of the system design procedure (and also part of many organisations' overall management of data). When an organisation develops such a model before it evaluates a package it obtains an invaluable tool for the package selection process. The model enables the organisation to compare the data structures it requires with the data structures the package caters for.

EVALUATING APPLICATIONS PACKAGES

In this section we examine the factors that an organisation needs to take into account when evaluating applications packages. Also, we consider the respective roles both of end users and of systems staff in the evaluation process.

When an organisation evaluates a package, it needs to take account of the following five categories of factors:

- Business-system-related factors.
- Operational factors.
- Historical factors.
- Supplier-related factors.
- Cost-related factors.

We discuss each of these categories in detail below, and, for each category of factors, we set out a checklist of questions an organisation needs to consider.

Business-system-related factors

Before an organisation undertakes the evaluation of potential applications packages, it will have identified and agreed the essential and the desirable objectives of the proposed system, and it will also have established the functional requirements of the proposed system. The organisation will evaluate packages against those requirements, and the evaluation should consider the different aspects of the package under the six headings of functionality, capacity, flexibility, usability, reliability and security. We set out below the questions that need to be answered under those six headings.

1. Functionality of the package
   - Do the inputs, the files and the outputs that the package provides include the required data elements, and does the package permit the required relationships to exist between those elements?
   - Does the package allow the required processing steps and the required calculations to be performed?
   - Does the package provide acceptable facilities to enable the end user to make ad hoc enquiries?
2. Capacity of the package
   - Are the file sizes adequate for the expected volume of data?
   - Are there unacceptable or risky limitations either on the number of transactions the package can process, or on the extent of table entries, or on the number of occurrences of a data item?
   - Does the package impose any unacceptable limitations of either response time or turn-round time?
3. Flexibility of the package
   - How easy is it to change the content of transactions, screen layouts and report formats?
   - How easy is it to either add or modify processing steps?
   - Can the package be used for other applications?
     (For example, several accounting packages that are sold, for marketing reasons, as a purchase ledger package, can be used as a sales ledger package as well.)

4. Usability of the package

   - Does the level of technical knowledge the end user requires match the level of expertise available? (This question may be particularly relevant with some modelling packages.)
   - Is the documentation complete and easy to understand?
   - Is the necessary training available for end-user staff?

5. Reliability of the package

   - Is the design of the package clear and modular, and is it capable of being developed further without the risk of creating undue problems of reliability?
   - Has the package been in use long enough and in a sufficient variety of circumstances for all of the errors in it to have been eliminated?
   - What and where are the weakest links in the package, and under what circumstances (if any) might it be expected to fail?
   - What are the consequences if the package fails, and what facilities does the package provide for recovery from a failure?

6. Security of the package

   - What control procedures are built into the package, and how do they compare with the organisation's standards?
   - What procedures does the package provide to ensure there is adequate back-up when a failure occurs?
   - What checks does the package perform to ensure that unauthorised access cannot be had to the package, and to ensure that the package will not process unauthorised transactions?
   - What audit facilities does the package provide?

Operational factors

The following questions need to be answered when the operational aspects of the package are considered:

- Does the operating style that the package requires conform to the operating style of the organisation, or does it conflict with it?
- Is the package easy to run?
- Is the expected operational performance of the package adequate for the expected number of users and transactions, and also for the expected file sizes, etc?
- Are the hardware requirements of the package acceptable, and what impact will they have on other jobs?
- Are the software requirements of the package (including the operating system, the teleprocessing monitor, the database management system, etc.) acceptable?

- Does the package impose any unacceptable constraints on the future choice either of hardware configuration or operating system?
- Is the source code of the programs available?
- Are the programs written and documented in a manner that would make in-house maintenance practical (if it became unavoidable)?
- What is the quality of the maintenance support that will be provided?
- Are sets of test data available for testing the package (both before the package is purchased and when new versions are introduced)?
- What is the quality of the technical operational documentation provided with the package?

Historical factors

The following questions need to be answered when the history of the package is considered:

- What is the origin of the package, and what is its development history since its inception?
- What is the current operational status of the package?
- How many organisations use the package both in this country and worldwide?
- What types of organisations use the package, and which of them are most (and least) successful in their use of the package? Also, why are some types of organisation more successful than others in using the package.

Supplier-related factors

The following questions need to be answered when the supplier of the package is considered:

- What is the commercial status of the supplier, and what is the commercial status of the particular package?
- Are both the supplier and the package likely to be in existence for the whole of the expected life of the system, and is the supplier likely to be able to provide effective support for the package for the same period?
- What is the supplier's reputation in the marketplace as regards both the quality of the package itself and pre-sales and post-sales support?
- Is the supplier's standard contract acceptable? (The points that need to be examined when an organisation reviews the supplier's standard contract are discussed on pages 44 and 45.)

Cost-related factors

The following questions need to be answered when the cost of the package is considered:

- What are the expected costs of acquiring and installing the package, and of operating and maintaining it for the whole of its expected life span?

— How do those costs compare with the costs of the alternative methods that could be used for providing the system?

The task of asking and obtaining answers to the questions listed above is, at first sight, a formidable one. However, the applications packages market in Europe in 1981 is a comparatively immature one. This present status of the market enables an organisation to reduce considerably the burden of obtaining answers to all the questions above merely by first considering two selection criteria. These two criteria (which are discussed below) are the local sales and technical support the supplier provides, and the number of existing users that a package has.

1. The local sales and technical support the supplier of a package provides
   The existence of local support, and particularly technical support, is essential if the purchaser of a package is to avoid the task of maintaining (or being ready to maintain) the package for himself. Since one of the potential benefits of using a package is that it frees scarce staff from maintenance work, this criterion is clearly an important one.

2. The number of existing users of a package
   With few exceptions (and these are mainly system suppliers and large United States package suppliers), few organisations are either able or willing to make the investment necessary to design and construct a proper package. Many so-called packages originate as once-off implementations that have been modified for subsequent purchasers. Typically, the maintenance costs associated with this approach become unacceptable when there are about six purchasers of the package, and the revenue from these initial purchasers is used to fund the redesign and redevelopment of the system as a proper package. Therefore, any package that has fewer than six purchasers (except for the packages supplied by the system suppliers and the large United States package suppliers mentioned above) is most unlikely to withstand close examination.

An example from the United Kingdom marketplace illustrates how the consideration of these two selection criteria can dramatically reduce the potential range of packages that an organisation needs to consider in detail. The example concerns the selection of a purchase ledger applications package that will be run on an IBM mainframe computer.

There are 99 different purchase ledger packages supplied in the United Kingdom by independent suppliers, of which only 18 are designed to run on an IBM mainframe computer. Of those 18 packages, 13 have sales and technical support available in the United Kingdom, but only five of those 13 packages have five or more users in the United Kingdom. This example illustrates how the number of packages that are worthwhile evaluating in detail can be reduced to a small number by applying just two selection criteria (assuming that the package is required to run only on one particular range of computer hardware).

The same result would apply in the above example if the package was required to run on Digital Equipment Corporation's PDP 11 range of computers rather than on an IBM mainframe computer. For any other range of computer equipment the number of potential packages that would be worthwhile evaluating in detail would be fewer than five.

Similar reductions in the number of packages that need to be considered in detail apply when the criteria in the example above are applied to other applications areas.

THE ROLES OF END-USER STAFF AND SYSTEMS STAFF IN THE EVALUATION OF PACKAGES

Our interviews with Foundation members lead us to conclude that the attitudes of end users towards the use of packages are influenced strongly by the attitudes of the organisation's systems development staff. In turn, the attitudes of systems development staff towards the use

of packages seem to depend primarily on the way they perceive their own role. If systems development staff perceive that their role is to provide the organisation with more effective systems for the minimum of effort and cost, then they approach the evaluation and selection of packages in an open-minded fashion.

If, on the other hand, systems development staff perceive that their principal role is to design systems, then they tend to devalue the potential usefulness of packages. In addition, computer operations staff and maintenance programmers tend to feel that the use of packages that do not conform to installation methods and standards complicates their job in an unwarranted way.

Because the backlog of development work is growing in most organisations, the use of a package, where it is appropriate, ought to be sensible both from the end users' point of view and from the data processing department's point of view. We say this even allowing for the possible adverse attitudes of the organisation's data processing staff.

Our interviews with package suppliers indicated that they prefer to deal directly with the end user, rather than with the data processing department. We were told of several cases where end users had undertaken the selection of a package without the professional advice of the data processing function, and in each case the end users regretted later that they had not sought this professional advice. The most common reason given for this regret was that the poor operational performance of the package led to excessive charges for machine resources.

In our view, the evaluation of packages ought to be carried out jointly by systems development staff and end-user staff. Also, the evaluation ought to be made within a formal systems development project management framework, which we discuss below.

When the requirements have been established, the systems development staff working on the project should conduct a preliminary screening of the market using the two selection criteria discussed on page 42. After this preliminary screening, the project team (consisting of end users and data processing professionals) will normally need to evaluate in detail no more than five packages. This detailed evaluation should be conducted on a project team basis, with each specialist undertaking part of the work. The tasks involved in evaluating a package compare with those necessary in any system development project and they include the following:

- The end user should assess the functional suitability of the package.
- The systems analyst should ensure that the capacity, the flexibility, the usability, the reliability and the security of the package are acceptable.
- The operations department should ensure that the package is acceptable from its viewpoint, and, together with the systems analyst, should advise the end user on the likely costs of running the package.
- Other concerned parties, such as the internal auditors and the external auditors, should ensure that any aspects of the package that are of concern to them are acceptable.

During our research for this report we did not discover any situations where the prospective purchaser had had difficulty in deciding which package to purchase once the detailed evaluation of the shortlisted packages had been completed. No organisation that we are aware of has found it necessary to develop a formal methodology (including relevant standards) for comparing packages.

PURCHASING APPLICATIONS PACKAGES

When an organisation is evaluating in detail a selected shortlist of packages, each supplier will submit a copy of his standard contract. As with all standard contracts, the prospective

suppliers' contracts will all be biased in their favour. Having decided on a package, the organisation needs to review closely the particular supplier's contract and to negotiate any necessary revisions to it before concluding an agreement to purchase that package from that supplier.

In reviewing the supplier's contract an organisation should normally examine in particular the following points:

- The scope of the agreement. The terms and conditions written in a contract supersede those given in any previous oral and written agreements or promises. All such agreements and promises that the supplier has made ought, therefore, to be incorporated into the contract.

- Delivery of the package. If the package needs to be modified before it is delivered, then the contract should specify a definite delivery date. If the organisation cannot tolerate a delay beyond the specified date, then the contract should provide for the purchaser to be able to cancel the agreement without penalty. The contract should specify the exact form and quantity of the items that will be delivered. In particular, each item of documentation that is to be provided should be explicitly stated.

- Acceptance testing of the package. If a suitable acceptance test is available or can be devised, its details should be included in the contract. If the only practical means of testing the package is for the purchaser to use it, then the contract should permit the purchaser to retain the package for a specified limited period of time without incurring any cost or penalty if the package eventually proves to be unsuitable.

- Warranty clauses. The warranty clauses in many contracts rarely provide much protection for the purchaser. Most suppliers state that they are not liable for consequential loss or damage, etc. While these clauses may not be defensible in a court of law, the purchaser can avoid the need for court action if, for example, the supplier's performance claims and the specifications of the package are included in the contract.

- Correction of faults. The time period during which the supplier will guarantee to correct all faults at no cost to the purchaser should be explicitly stated in the contract. Ideally, this guarantee should be in force for all or for most of the life of the package. If possible, the purchaser should also obtain a guarantee of the time that the supplier will take to fix a fault. If the purchaser obtains such a guarantee, he needs to ensure that it is reinforced with a non-performance penalty. If the package is to be used in an interactive environment (particularly for transaction input routines), then this issue becomes even more important than it would be if the package was intended to be used in a batch processing system.

- Modifications to the package. If the package is to be modified before it is delivered, then the changes need to be specified in detail in the contract. In addition, the contract should include details of the procedures that will be used to accept the changes, and it should specify the delivery deadline and the cost (if any) to the purchaser. Also, the contract should include an agreement that the supplier will carry the modifications forward to any later versions of the package.

- Functional and operational enhancements to the package. Many suppliers improve their products to provide new facilities and to accommodate equipment and systems software developments. The conditions under which the purchaser can obtain these enhancements ought to be specified in the contract.

- Modifications to the package that the purchaser makes. The purchaser's rights to

modify the package should be specified, together with the effect this will have on the support that the supplier will provide. In general, however, any changes the purchaser makes will invalidate the guarantees and warranties the supplier has given.

- Optional features. The package may have optional features that the purchaser may wish to acquire at a later date. If it does, then the purchaser should attempt to negotiate favourable conditions for purchasing those features. The purchaser should also ensure that these conditions are included in the contract.

- Multiple implementations of the package. The purchaser may well wish to install the package on more than one computer (particularly in a distributed processing environment). The contract should include details of the incremental cost to the purchaser of installing the package on additional computers.

- Quality of the supplier's personnel. The supplier may provide training, modification support, or other services. The qualifications of the individuals concerned should be explicitly stated in the contract, and the purchaser's right to refuse the services of specific individuals should be protected by an appropriate clause in the contract.

- Terms of the lease. If the purchaser leases the package, then the contract should contain a clause that gives the purchaser the right to renew the lease, and ideally this clause should specify the renewal terms. The contract should also specify the conditions under which the purchaser can cancel the lease.

- Schedule of payments. The initial payment should be contingent on the delivery of the package, on the delivery and acceptance of any modifications, on the receipt of the attendant services (such as training), on the completion of acceptance testing, and on the successful installation of the package.

- Title to the package. The contract ought to state either that the supplier has the title to the package or that he has the marketing rights to it. The purchaser should ensure that the contract states that the purchaser will not be held liable if subsequently the supplier is found to have misrepresented this information.

- Laws governing the agreement. Most contracts contain a clause that specifies the country in which any court action concerning the contract will take place, and under whose laws the action will be decided. Normally, this will be the country or the state in which the supplier has his main office. The purchaser should attempt to change this clause so that it specifies the purchaser's country.

In reality, there are few, if any, suppliers who are likely to agree to incorporate all the points mentioned above. Indeed, an organisation would be unwise to do business with a supplier that did so. Nevertheless, given that the starting point for negotiations will be the supplier's standard contract, the purchaser should be able to obtain a better deal by negotiating some or all of the above points.

SUMMARY

In this chapter we have shown that the task of selecting and installing applications packages should take place within the overall project management structure for systems development. These tasks should not be left for end users to perform by themselves. Instead, they should be carried out by a project team comprising specialists drawn from end users, systems development staff, computer operations staff and certain other interested parties (such as internal auditors and external auditors).

The advantages of reducing both the cost and the time taken to develop a system using applications packages can be so great that end users are sometimes prepared to compromise on their requirements in order to use a package. We believe, therefore, that when an organisation is specifying the requirements of a system, it is of paramount importance to distinguish between those features that are essential and those that are merely desirable. If this distinction is not made, then it is almost certain that the organisation will not be able to find any package that meets all of the requirements. On the other hand, if packages are evaluated against only the essential requirements of a system, then there is a greater probability that a suitable package will be found. In those circumstances, the end user may be prepared to forgo the desirable (but inessential) requirements in order to gain the advantages of using that package.

We have also listed in this chapter the questions that an organisation needs to consider when evaluating an applications package. The list of questions appears, at first sight, to be formidable, but we have shown how the application of a few of the selection criteria can substantially reduce the number of packages an organisation needs to consider in detail.

Finally, we have listed the contractual points an organisation needs to consider before concluding an agreement to purchase an applications package.

CHAPTER 6

THE MANAGEMENT IMPLICATIONS OF APPLICATIONS PACKAGES

Ten years ago, applications packages were of little strategic importance to those responsible for managing an organisation's information processing function. Today, most of the managers concerned accept that applications packages have an important role to play in the development of systems. This change in attitude has come about because the rapidly expanding demand for computer-based systems generally cannot, in most organisations, be fulfilled by developing in-house bespoke systems. The increased demand for systems has been brought about primarily by the continuing, and dramatic, improvements in the price/performance ratio of computer hardware. In contrast to the many improvements that have been made in the productivity of hardware, virtually no improvements have been made in the productivity of providing applications software. Consequently, many organisations now find that they have to devote more software development resources to maintaining those systems that they have already implemented than they devote to developing systems to meet new requirements.

Applications packages, in the right circumstances, have the potential to improve the overall productivity of systems, and for many applications systems they can provide a valid alternative to bespoke systems. In this chapter, we review the management implications of using applications packages. We begin with a review of the pressures that are causing the increased demand for applications systems, and we show how applications packages can help to relieve those pressures and meet the demand. We then examine the way in which the applications packages marketplace is likely to develop in the future. Finally, we set out the management issues that an organisation needs to consider if it is to make better use of applications packages.

THE INCREASING DEMAND FOR APPLICATIONS SYSTEMS

The demand for applications systems is growing rapidly, and there are several reasons for this growth (the most important of which are discussed below). In addition to the growth of the number of computers installed, there are two other factors that explain the growth in the demand for applications systems.

The first factor concerns the growth in the backlog of development work. A survey conducted in the United States during 1979 indicated that more than 90 per cent of the organisations that have installed a mainframe computer in that country have a schedule of development work reaching at least three years ahead. Also, a recent survey of European computer installations has shown that the backlog of applications development work is now the primary matter of concern of many information systems managers.

The second factor that explains the growth in the demand for applications systems concerns the shortage of skilled staff. A recent study sponsored by the British Government (*Computer manpower in the '80s*) concluded that there is, at present, a shortage of at least 16,000 programmers in the United Kingdom. The study predicts that to sustain even the most conservative rate of growth for applications systems, a minimum of 6,000 additional analysts and programmers will be required each year until at least 1985.

In addition to the increased demand for computer applications generally, there are changes also in the pattern of demand for certain types of applications. The two most significant trends are:

- An emerging demand for applications that can collectively be described as "personal computing". Microcomputers will increasingly be used for personal computing, and as we mentioned in chapter 2 (on page 10), microcomputer packages form a fast-growing branch of the applications packages market. Executives will use microcomputer personal computing applications to construct and use computer systems to assist them in analysing information and making decisions.

- A growing demand for business communications applications such as word processing and electronic mail.

Also, changes have taken place in the systems strategy in many organisations, and those changes have increased the demand for applications systems. The main change is the trend towards using distributed computer facilities. Distributed computers either create the need to use computer applications in parts of the organisation that have not previously had the use of computer-based systems or, alternatively, create the need to develop or acquire replacement applications systems that are designed to operate in a distributed processing environment.

THE USE OF APPLICATIONS PACKAGES TO MEET THE INCREASING DEMAND

The increasing demand for applications systems that we identified in the previous section can be met either by deploying more systems development resources or by improving the productivity of the existing systems development staff. Deploying more development resources is (as we found in the survey we carried out for this report) a remedy that Foundation members as a whole do not find appealing. In any case, this remedy could not be applied universally, because the required staff neither are, nor are likely to be, available in either the short-term or the medium-term future.

It follows, then, that the best way of meeting the increased demand for applications systems is to improve the productivity of the existing systems development staff. The use of applications packages is one of the several ways in which this productivity can be improved. However, before an organisation seeks to improve the productivity of its systems development staff, it needs to be clear about the wider issue of the productivity of systems. In Foundation Report No. 11 — Improving Systems' Productivity, we said that the objective of the systems development function is to deliver the optimum systems capability to the end users, so that they, in turn, can optimise the organisation's productivity. From this point of view, the role that applications packages have to play in improving the productivity of systems development staff is potentially much wider than it might at first appear to be.

In our view, there are powerful reasons why an organisation should make greater use of applications packages. The five most important are:

- Scarce analysts and programmers are released to work on those applications that are unique or special to their organisation.

- The same scarce resources are released from maintenance work, since most, or all, of this work can be subcontracted to the package supplier.

- The systems development department can deploy more resources on developing new systems. By doing this, it enhances its overall standing in the eyes of the rest of the organisation, since it can be seen to be responding to end users' demands for new systems.

- The best packages have a sufficient number of users to ensure that they are well supported, that they meet the needs of many different end users, and that they will endure even though the needs of the business may change.

— Applications packages are available more quickly than are bespoke systems, and they provide software that is more reliable than brand-new software.

The reasons listed above make it worthwhile, in our view, for an organisation to consider always whether applications packages can be used as part of a systems development project.

On the other hand, there are three reasons that an organisation might give for rejecting the use of applications packages, and we briefly discuss these reasons below:

1. Matching a package with the perceived needs
   A particular applications package is unlikely to meet all of an organisation's perceived needs. During the research for this report, however, we detected that organisations were increasingly willing to compromise on this issue. In the current economic climate, this willingness to compromise stems partly from the attraction of the cheaper solution that a package offers as compared with the cost of developing a bespoke system. It stems also partly from a recognition that the use of a package may be the only way of implementing a system reasonably quickly, given the typical backlog of development work that most organisations now have.

   Also, the packages available today are more flexible than packages were in the past, and, if the perceived needs are reduced to the essential requirements, an organisation is more likely to find a package that meets its perceived needs.

2. Conflict with other elements of an organisation's systems strategy
   Some elements of an organisations's hardware and software strategy may make it less feasible to use applications packages. For obvious commercial reasons, independent package suppliers develop their products with the largest possible market in mind. Thus, from the independent suppliers' point of view, the most attractive segment of the market is that provided by an IBM 4300/DOS/CICS/VSAM environment. There are, of course, some independent suppliers of packaged software that provide products that can be used with many other manufacturers' hardware and software. But those organisations that have a non-standard systems environment will have less opportunity to use packages than will those organisations that use standard hardware and software products (irrespective of whether those products are IBM's or any other manufacturer's).

   While the cost of developing bespoke applications software continues to increase, organisations will, in our view, increasingly need to consider whether other elements of their systems strategy reduce their options for using applications packages. In particular, an organisation may need to forgo the use of desirable, but non-standard, software such as teleprocessing monitors and database management systems. When an organisation decides whether or not to use such software, it needs to consider the impact the software will have on the organisation's choice of applications packages. The implication is that the organisation should use either the standard software products of the system supplier, or the software products of the market leader amongst the independent suppliers.

3. Other means for improving the productivity of systems
   Several techniques and methods for improving the productivity of systems have been and still are being developed. Some of these techniques and methods were identified in Foundation Report No. 11, and they and certain others will be examined in more detail in Foundation Report No. 25. The effect of these developments may be to diminish, in the eyes of some organisations, the importance of applications packages as a method for improving the productivity of systems. The developing techniques and methods will undoubtedly improve the productivity of individual areas of the systems development process. Despite this, we do not believe that they herald a major breakthrough that will have a dramatic impact on the life-cycle costs of systems. In our view, the use of applications

packages will be an important method of improving the productivity of systems for the foreseeable future.

A final important aspect of the decision on whether or not to use applications packages concerns the end users' attitudes and preferences. Once end users have had the experience of using packages that are professionally designed, professionally marketed and professionally supported, they may well prefer in the future to use packaged software, rather than software designed and constructed to their own specification. There are two main reasons why we believe that end users may prefer to use packaged software:

- A package is a tangible product that does not require the considerable amount of abstract thought that is required to design a bespoke system. There is, therefore, less risk that the system specification will prove to be either inaccurate or inappropriate.

- A package, and in particular a package that has a substantial customer base, is more likely to be able to cater for either unforeseen needs or changed needs than is a bespoke system.

THE FUTURE OF THE APPLICATIONS PACKAGES MARKETPLACE

In chapter 2 of this report, we highlighted the major trends to date in the applications packages marketplace. In this section we now project those trends into the future, and we highlight the major trends as we see them.

A small number of independent software suppliers will emerge whose principal business it will be to develop, market and support applications packages. Most of those suppliers will be from the United States, but they will see Europe as a potentially large market for their products. They will concentrate on supplying comparatively expensive applications packages (in the £30,000 to £45,000 price range), that they will design and market for use with mainframe data processing applications (in particular, financial and manufacturing applications).

Because of their substantial customer bases, those comparatively few suppliers will have the financial resources, the management resources, the technical resources and the experience to provide significantly better applications packages than other organisations can provide. Their products and marketing strategies can be expected to:

- Concentrate on standard products, with no modifications provided for individual users.

- Concentrate on standard hardware and software environments, and in particular on those provided by the market leaders.

- Base their products on database concepts, so that those products can be integrated into an organisation's overall data management strategy.

- Provide facilities such as high-level languages and simple links to other applications, so that the packages can be used as the basic building blocks to form a large proportion of the systems that, overall, will meet nearly all of the end users' needs.

- Concentrate their marketing effort on end-user management, rather than on data processing management.

Data processing system suppliers will also become increasingly active in the applications software market, and there are two reasons for this. First, applications software is a fast-growing market that provides them with an opportunity for increased revenue and profits. All system

suppliers are already active in the applications software market in some form, and so they already have a technical base, a product base and a marketing base on which they can build.

Second, the availability of good packaged applications software will, in the future, be a prerequisite for increased hardware sales, because the cost of developing bespoke applications will increasingly make further computerisation uneconomic. Some observers might argue also that if the system suppliers provide the applications software themselves they will be able at least to maintain, and perhaps even to increase, their influence on the overall productivity of systems. By so doing, they will obviously enhance their prospects for achieving additional hardware sales.

So far, we have restricted our discussion of the future of the applications packages marketplace to those packages that are designed for use on mainframe computers. The past few years have, however, seen the emergence of both the minicomputer and the microcomputer in several different guises. The main growth in the number of available software products is for packages designed to run on these two types of computer. Nevertheless, these packages are (with a few exceptions) not comparable, either as regards the facilities they offer or their cost, with the packages that are provided for use with mainframe computers. The market for minicomputer and microcomputer software is one in which the supplier has to seek high-volume sales at a low unit cost. Consequently, it is a more risky market than is the market for mainframe computer software. There are four main reasons for the differences between the two markets:

- There are many more types of minicomputer than there are types of mainframe computer.
- The price of the software needs to be in proportion to the cost of the hardware. Minicomputers cost substantially less than mainframe computers, and their cost is continuing to fall.
- The cost of marketing applications software for minicomputers is comparable to the cost of marketing applications software for mainframe computers, but the potential revenue per package sold is lower.
- Given the inadequate legal protection that is afforded to the copyright of software, there are greater risks of the unauthorised use of minicomputer packages than of the unauthorised use of mainframe computer packages.

From these differences between the two types of market, we conclude that good applications packages both for minicomputers and microcomputers will emerge from the independent software suppliers only in very specific cases. One example is where a package developed for a more expensive computer is "downward" compatible.

A more likely trend both for minicomputer packages and for microcomputer packages is illustrated by the existing word processing sector of the systems marketplace. A few people already recognise word processors for what they are — a bundled package of hardware and applications software. We expect the combination of standard applications software and suitable hardware to be an increasingly important part of the small systems marketplace. A combined hardware and software package will become the principal way in which such systems can be provided both in sufficient numbers and with a sufficient profit margin to justify suppliers taking the necessary risk and investment.

In summary, the market for applications packages can be expected to diverge in the following ways:

- In the mainframe computer market, applications packages will remain unbundled, and they will become increasingly sophisticated, with the major system suppliers joining the

handful of successful independent software suppliers for a share of the market. The independent software suppliers will concentrate on providing packages for use with the hardware of the leading system suppliers, and this will increasingly create an area of confusing commercial relationships. The independent suppliers will be both allies and opponents of the leading system suppliers.

— In the minicomputer and the microcomputer markets, the emphasis will be on bundling hardware and software together to provide a "systems solution" for a specific application. For the reasons outlined earlier in this section, the products available in this sector of the market, and the way in which the suppliers (and especially the independent suppliers) market and support them will be inferior to the products, the marketing and the support available in the mainframe computer sector of the market.

MAKING BETTER USE OF APPLICATIONS PACKAGES

Those responsible for managing an organisation's systems development function now generally accept that applications packages will form an increasingly important part of their toolkit for developing systems. In this final section of the report we set out the six actions that we believe the management of the systems development function of an organisation should take to ensure that the organisation evaluates and uses applications packages in the most effective way.

1. Applications packages and the selection of hardware and software
   When an organisation is planning significant hardware and software purchases, it needs to consider the impact those purchases will have on the choice of applications packages. In particular, management should ensure that the planning staff consider the possible effects that the purchase of non-standard equipment or software will have on the choice of applications packages.

2. Applications packages and project management procedures
   The organisation's project management procedures and codes of practice should be reviewed to ensure that packages are considered properly wherever they may be applicable. In particular, the feasibility stage of a system development project should include an explicit consideration of the trade-offs between the value of the required benefits and the cost of providing those benefits.

3. The skills required to evaluate and purchase applications packages
   The management of an organisation's systems development function should recognise that a special set of skills are required to evaluate packages, to negotiate contract terms and to install packages. These skills should be concentrated either in one individual or in a small group working in a part of the business (such as the financial accounts department) that commonly uses packages.

4. The attitudes of systems development staff towards applications packages
   Systems development staff should be encouraged to develop the correct attitude towards the productivity of systems. They should derive as much satisfaction from selecting and installing purchased software as they do from designing and constructing their own bespoke systems. Systems development staff will develop this attitude only if they recognise and accept that their main aim should not be to improve their own skills and experience in designing systems, but rather to provide the required systems to the end users.

5. The cost of developing in-house packages
   Any proposals put forward for developing an in-house package should be questioned very closely. There are undoubtedly several emerging applications (notably in the field of office automation) where the number of separate implementations of the software would justify

the considerable cost of developing an in-house package. But, unless adequate resources can be made available for developing an in-house package, it would almost certainly be better either to develop an individual system for each end user or to purchase an externally-developed package. Of those two alternatives, purchasing an externally-developed package would probably be the more cost-effective solution, particularly when the costs of maintaining the package are also taken into account.

6. Applications packages and end users
   To make the best use of applications packages, end users need to place somewhat less emphasis on the uniqueness of their own requirements. They need instead to place more emphasis on the similarities that exist between their own needs and the needs of other organisations. However, those responsible for managing the systems development function should ensure that end users are aware of the pitfalls that can occur when packages are evaluated and purchased. As we pointed out on page 50, package suppliers will increasingly concentrate their marketing effort on end-user managers. Steps need to be taken, therefore, to make end users aware that when they are evaluating and purchasing packages, they need to seek the advice of professional systems development experts.

The management actions that we have listed above are not meant to be exhaustive. But they include the actions that we believe an organisation should take if it is to make the most effective use of applications packages.

All the user organisations that we surveyed or interviewed during the research for this report are concerned about the productivity of their systems development staff, and also about their ability to satisfy the end users' demands for systems. In this report we have shown that applications packages have the potential both to improve the productivity of systems development staff and to satisfy the demand for new systems.

The main conclusion that we have come to in this report is that many systems development departments will be well advised to place more emphasis on the use of applications packages. If they do that, they will be able to make a significant improvement in the service that they provide to their end users.

Photoset and printed in England by Flexiprint Ltd, Worthing, Sussex

The Butler Cox Foundation

Report Series No 22 Applications Packages

Please record your interest in follow-up work on the subject of this report, highlighting those aspects of the topic of most interest to your organisation.

Completed by.....

Job title.....

Organisation.....

Address.....

.....

Please return to: The Butler Cox Foundation
Butler Cox & Partners Limited
Morley House
26 Holborn Viaduct
London EC1A 2BP

The Butler Cox Foundation

Report Series No 22 Applications Packages

Please note below any points on which you would like the author to expand at the professional and technical seminar associated with this report.

In order to take account of your comments we need to receive them from you by 16 March 1981.

Completed by.....

Job title.....

Organisation.....

Address.....

.....

Please return to: The Butler Cox Foundation
Butler Cox & Partners Limited
Morley House
26 Holborn Viaduct
London EC1A 2BP

Butler Cox & Partners Limited
Morley House, 26-30 Holborn Viaduct, London EC1A 2BP
Tel 01-583 9381, Telex 8813717 GARFLD

Butler Cox & Partners Limited
216 Cooper Center, Pennsauken, New Jersey 08109, USA
Tel (609) 665 3210

Italy
Sisdoconsult
20123 Milano – Via Caradosso 7 – Italy
Tel 86.53.55/87.62.27

Benelux, Germany, Switzerland
Akzo Systems bv
Velperweg 76, Arnhem, The Netherlands
Tel 85-662629

France
Akzo Systems France
Tour Akzo, 164 Rue Ambroise Croizat,
93204 St Denis-Cedex 1, France
Tel (1) 820.61.64

The Nordic Region
Statskonsult
PO Box 4040, S-171 04 Solna, Sweden
Tel 08-730 03 00, Telex 127 54 SINTAB-S