Management Summary

BUTLER COX FOUNDATION

Managing Multivendor Environments

BUTLER COX
FOUNDATION

Managing Multivendor Environments

Management Summary
Report 72, November 1989

Butler Cox plc

LONDON
AMSTERDAM MUNICH NEW YORK PARIS

Published by Butler Cox plc
Butler Cox House
12 Bloomsbury Square
London WC1A 2LL
England

Availability of reports

Members of the Butler Cox Foundation receive three copies of each report upon publication; additional copies and copies of earlier reports may be purchased by members from Butler Cox.

Photoset and printed in Great Britain by Flexiprint Ltd., Lancing, Sussex.

Managing Multivendor Environments

Management Summary

This document summarises the main management messages from Foundation Report 72, published in November 1989. The full report is available to members of the Butler Cox Foundation.

Until recently, many organisations used equipment provided by a single vendor — often IBM. Others based their computer systems on IBM's architectures, arguing that there was sufficient competition amongst the IBM plug-compatible suppliers.

Today, the situation is different. First, the range of products and IT applications has expanded faster than IBM's capability. Few now believe that IBM can, or will be able to, provide solutions to all computing problems, and some very strong 'niche' vendors have emerged to exploit the gaps. Second, the increasing trend for computing responsibility to be devolved to business units means that business managers now have greater freedom of choice about the computing products they use.

For both of these reasons, a multivendor computing environment is inevitable for most large organisations.

Indeed, for many years, it has been common to use different vendors' computing environments for scientific/technical computing and for mainstream data processing. Apart from the additional costs of separate teams of systems staff for each environment, the problems of managing a multivendor environment have not presented any great technical difficulties while applications running in the different vendors' environments have operated independently of each other.

However, the problems, and the costs, of a multivendor environment escalate when the products from one vendor need to interwork (or be integrated) with the products of another vendor. The need for such integration is increasing and is often unforeseen. We know of many organisations that are faced with the problems of integrating applications that were developed independently of each other and that have hitherto been completely separate.

Many Foundation members now have a multivendor environment

Systems directors must therefore find the right balance between choosing the best hardware and systems software for each application, and increasing the cost and technical complexity that arises from having to integrate applications running in the different vendors' environments. To get this balance right, they need to understand the cost implications of a multivendor environment, and to concentrate their efforts on those areas where the impacts are greatest.

The distinction between multivendor and multi-architecture environments is important

Multivendor environments occur in two situations:

— Where equipment and software from different vendors are used to construct a computer system that conforms to a *particular software environment.* This situation is commonly found with IBM plug-compatible hardware that conforms to IBM's proprietary architecture. We refer to this as a single-architecture environment.

- Where applications running in *different software environments* (usually defined by two different manufacturers' proprietary architectures) need to be integrated. We refer to this as a multi-architecture environment.

Figure 1 shows that there are advantages and disadvantages associated with each of the four possible combinations of single/multi-architecture and single/multivendor. Organisations may deliberately position themselves in a particular quadrant or, indeed, choose to move from one to another, to gain particular advantages:

- A single architecture, single-vendor environment will provide technical and managerial simplicity.
- A single-architecture, multivendor environment will increase the negotiating power of an organisation because it will no longer be dependent on a single supplier. Thus, users of IBM and plug-compatible machines have found that they can make large savings, and users of Unix systems find that the hardware is priced very competitively.
- A multi-architecture, single-vendor environment will make for a simpler relationship with the vendor, although integrating applications in the different environments will not be straightforward.
- A multi-architecture, multivendor environment will give an organisation the greatest flexibility to select the best combination of hardware and software for a particular application, regardless of which vendors supply them.

Figure 1 It is important to distinguish between multivendor and multi-architecture environments

	Single vendor	Multivendor
Multi-architecture	*Examples*: IBM: MVS and DOS; Bull: GCOS6, GCOS7, and GCOS8 *Advantage*: Simpler relationships with vendors *Disadvantages*: Limited ability to choose most appropriate functionality for applications; Reliance on a single vendor	*Examples*: ICL and IBM; IBM and Digital *Advantages*: Wider range from which to choose best product for the job; Reduced dependence on a single vendor; Rapid access to new technology *Disadvantages*: Limited multisourcing; Increased complexity
Single architecture	*Example*: Siemens BS2000 *Advantage*: Simplicity *Disadvantages*: Reliance on a single vendor; Reduced ability to choose most appropriate environment for a particular application	*Examples*: IBM and Amdahl; MS-DOS *Advantages*: Negotiating power; Reduced dependence on a single vendor; Opportunities for multisourcing *Disadvantage*: Reduced ability to choose most appropriate environment for a particular application

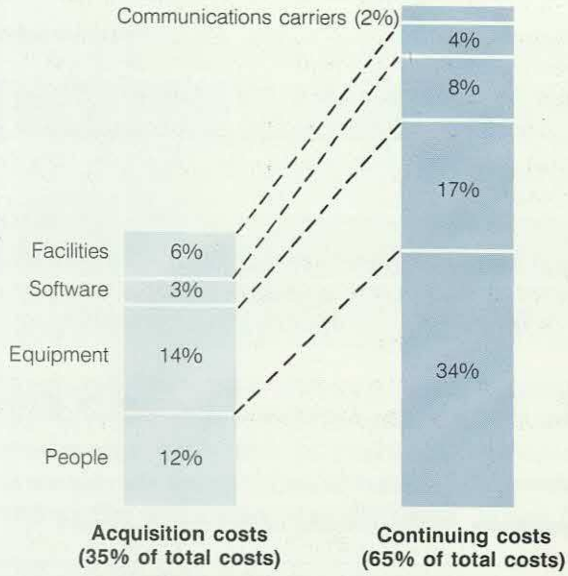
Multi-architecture, multivendor environments usually cost more than other equivalent environments

During the research, we found that many Foundation members were unclear about the cost implications of a multivendor environment, compared with a single-architecture, single-vendor environment that provides comparable facilities and performance. Most had recognised that the initial costs of acquiring hardware and systems software can be lower in a multivendor environment, but had not recognised that additional ongoing costs, incurred over several years, could outweigh the initial savings.

A breakdown of the total cost of owning and running a typical single-architecture, single vendor mainframe environment is shown in Figure 2. As the figure shows, over a five-year period, about two-thirds of total costs are attributable to continuing costs. Continuing staff costs (34 per cent) are by far the largest cost item, followed by continuing equipment costs (17 per cent), and initial equipment-acquisition costs (14 per cent).

Figure 2 The total cost of owning and running a computer facility comprises acquisition costs and continuing costs

Over a five-year period, continuing costs account for two-thirds of the total costs of owning and running a typical computer facility; continuing personnel costs are the single largest item of expenditure.



In a single-architecture, multivendor environment, the main area in which savings can be achieved is in hardware-acquisition, because it is possible to negotiate the best deal from competing vendors. Typical savings of between 10 per cent and 40 per cent can be expected. These savings will be partly offset by higher software licence fees, and slightly increased continuing staff costs.

In a multi-architecture, multivendor environment, cost savings can be achieved not only in the acquisition of hardware, but also in applications software acquisition costs (because there is a greater choice of packages that can be used). These savings are likely to be more than outweighed, however, by the substantial additional costs, especially staff costs, required to integrate applications in the different software environments.

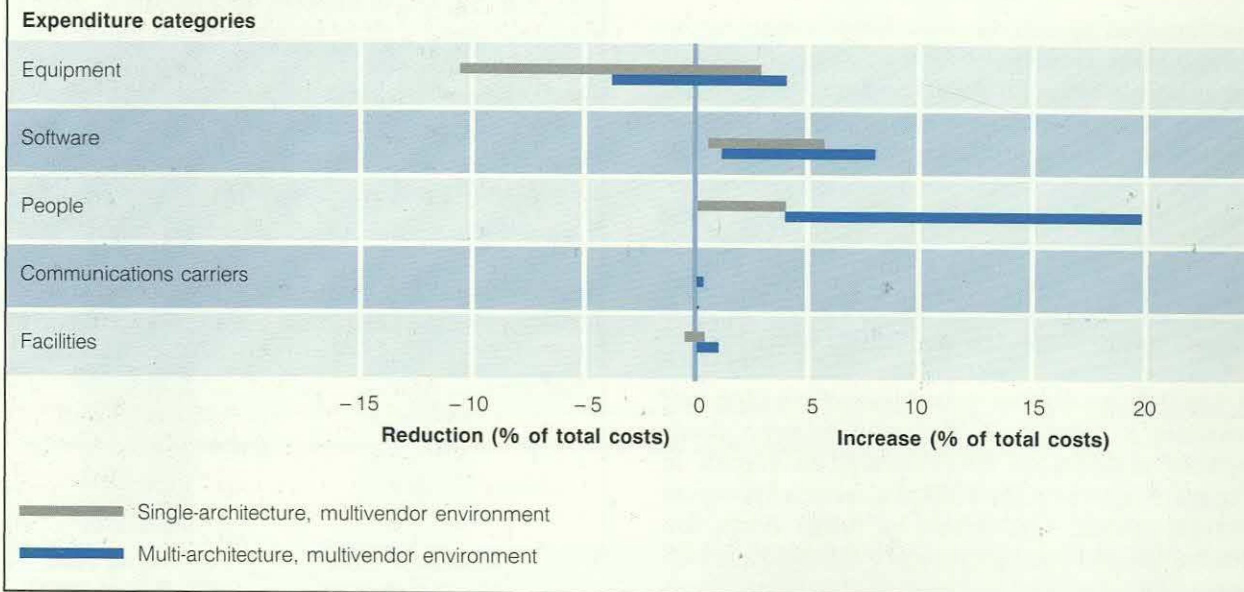
By combining the savings and additional costs attributable to a multivendor environment with the corresponding portions of the cost-of-ownership profile shown in Figure 2, it is possible to obtain an overall prediction of the additional costs and savings likely to result from a multivendor environment. Figure 3, overleaf, illustrates the results and shows clearly that, in a single-architecture, multivendor environment, equipment and software are the categories most likely to be affected. Reductions in expenditure on equipment could reduce the total cost of the systems function by more than 10 per cent, whereas additional expenditure on software could increase total costs by more than 5 per cent.

Figure 3 shows that in a multi-architecture, multivendor environment, expenditure on people is the area that will be most affected. The total costs of the systems function could be increased by as much as 20 per cent because of the additional costs of people. In contrast with the situation for a single-architecture, multivendor environment, reductions in total costs resulting from lower hardware expenditure will be insufficient to compensate for the extra expenditure on people.

Thus, the costs of a multi-architecture, multivendor environment are likely to be higher than those incurred in an equivalent single-architecture environment, regardless of whether it is single-vendor or multivendor. There may well be good business reasons for being in a

Figure 3 Increases or decreases in cost will vary in the different expenditure categories according to whether an organisation is operating in a single-architecture or multi-architecture, multivendor environment

The figure shows the cost implications of a multivendor mainframe environment compared with a single-vendor, single-architecture environment. For each type of multivendor environment, it shows the range of percentage savings or increases in the total systems costs that can be attributed to each of the five main expenditure categories. In a single-architecture, multivendor environment, the biggest effects will be savings on hardware expenditure and increases on software expenditure. In a multi-architecture, multivendor environment, the biggest effect is likely to be increases in the costs of people.



multi-architecture environment, but the systems director should ensure that the cost implications are considered before a deliberate choice is made to move to such an environment.

The cost implications of a multivendor environment, summarised in Figure 3, relate specifically to mainframe systems. Nevertheless, the same general pattern of cost savings and increases is likely to be found for multivendor minicomputer environments.

A well designed technical architecture will help to reduce integration problems

Integration problems arise in multi-architecture environments when it is necessary to interlink applications running in different software environments. In the remaining sections of this document, the term 'multivendor' also implies multi-architecture, although similar problems arise in a single-vendor, multi-architecture environment. It can be almost as difficult to integrate applications running in two IBM software environments as it is to integrate applications running, for example, in IBM and Digital environments.

The integration problems affect the overall costs of a multivendor environment, the timescales involved in developing new applications, and thus the quality of the service that the systems function provides to its users. The difficulties of sharing data between applications in different software environments, updating and enhancing such applications, and facilitating interworking between them, are all determined by the organisation's overall technical architecture.

The technical architecture provides the framework within which hardware and software will be acquired, and applications will be developed and run. It therefore defines the hardware architecture, the software infrastructure, and the applications architecture (see Figure 4). By defining a suitable technical architecture, the systems department can minimise integration problems.

The most significant element of the technical architecture is the software infrastructure. Foundation Report 69, *Software Strategy*, described in detail how to define and construct the software infrastructure. The software infrastructure should be designed to minimise both the number of different software environments, and the differences between them, so that as much as possible of the applications coding can be independent of the environment.

There is a general trend for software environments to be designed so they can be used with a wide range of machine sizes, and such environments should be chosen wherever possible. Digital's VMS operating system, for example, spans a wide range of machine sizes, and Unix is now a cost-effective environment for general-purpose workstations as well as for departmental computers. One of the aims of IBM's SAA is to broaden the range of machine sizes that can be covered with one software environment.

It may be possible to overcome the differences between different software environments by using software products and development tools designed to be used in more than one environment. Examples include the products and tools available from Oracle, and the Focus family of products available from Information Builders. By writing applications to conform to the 'standard' defined by such products and tools, the differences in the underlying operating systems can, to some extent, be hidden from the applications.

There will nearly always be a requirement for some sort of link between the different software environments, however, and as the need for integration continues to increase, these links will become more pervasive and complex. There are three different types of integration:

— Data-driven, where the same data is used by applications in different environments.

— User-driven, where the same users need to access applications running in different environments.

— Process-driven, where the same business process is supported by applications running in different environments.

Integration between software environments may be required at two different levels — batch interchange or online in realtime. Different elements of the technical architecture are critical for achieving the different types of

Figure 4 A technical architecture defines three main components

The technical architecture defines the hardware architecture, the software infrastructure, and the applications architecture. The level of detail depends on the extent to which systems activities are devolved to business units.

The *hardware architecture* defines principles and policies in the following areas:

— How many hardware environments will be used, and which suppliers should provide the different hardware elements.

— Which are the strategic vendors. They are likely to be those vendors whose products support long-term applications that have a significant impact on the business, because they would cost too much to replace, or because they produce large benefits, or because some business areas would not be able to function without them.

— The basic functions and uses of the different hardware environments. For example, one might be used for office automation, while another might be used to provide all other computing facilities.

— Where the various hardware elements are to be located, and who will be responsible for their day-to-day and long-term management.

— The network layout, both for wide-area communication between computer centres and user departments, and for local communications within each site.

— How different hardware environments are to be connected and which gateways or other interfaces will be used to link them.

The *software infrastructure* defines the operational software environment which runs on the hardware environment and in which applications will be supported. It is the central component of an organisation's software strategy, and is described in Report 69, *Software Strategy*. The main components of the software infrastructure are:

— The operating system and development environment.

— Data-management software.

— Communications software.

— User-interface standards and software.

— Core applications, which Report 69 defines as applications that usually maintain data used by more than one department.

The *applications architecture* defines which functions need to be provided, how the applications providing these functions relate to each other, and on which data structures they rely. The applications architecture therefore sets out:

— The core and non-core applications, and the functions they support.

— The mapping of applications onto the hardware architecture and the software infrastructure.

— The applications that need to exchange data.

— The main data structures relevant to the applications.

— The main users of different applications.

integration. Figure 5 summarises the critical elements for each type and also shows the means by which each combination of type and level can be implemented.

Standards are useful but they are not a panacea

The technical architecture will need to take account of the standards that will be used. At their present stage of development and application, standards are most useful for interlinking computer systems so that data can be transferred between them. X.25, Ethernet, SNA, and other standards are all useful in this respect, and hardware and software products to implement them are available for most software environments. The nature of the link required and the hardware will usually determine the most suitable choice of standards.

Standards, however, are not a complete answer to integrating applications in different software environments. The main difficulties are two-fold:

- There are often several versions of the same standard. The X.25 specification, for example, permits different implementations. Similar problems occur when a basic standard (SQL, for example) is extended and enhanced in different ways by different vendors, so that full compatibility across different software environments is unlikely to be achieved.
- In several areas, standards are not sufficiently widely adopted to facilitate easy integration. The lack of open network-management standards is a case in point.

The multiplicity of established and emerging standards means that the systems department needs to take the initiative itself in many areas:

- *Define its own standards policy:* To do this, the systems department will need to understand how standards are developing. Several organisations have 'standards watchers' whose role is to track the development of standards.
- *Decide how it can influence the evolution of standards:* The systems department should identify the areas in which standards are important to it, and include standards issues in evaluating tenders from competing vendors.
- *Specify in-house standards:* The in-house standards policy will refine the choice of external standards, by deciding which variants will, or will not, be used, and by setting in-house standards where there are no suitable external standards.

Considerable effort will be required to define and maintain the technical architecture

In general, the development of the technical architecture will require at least two to three man-years of effort, spread over a year's elapsed time. Further effort will be required to review the technical architecture, probably annually, once it has been implemented.

The skills required to define and maintain a technical architecture differ from normal

Figure 5 The different elements of the technical architecture are critical for the different types of integration

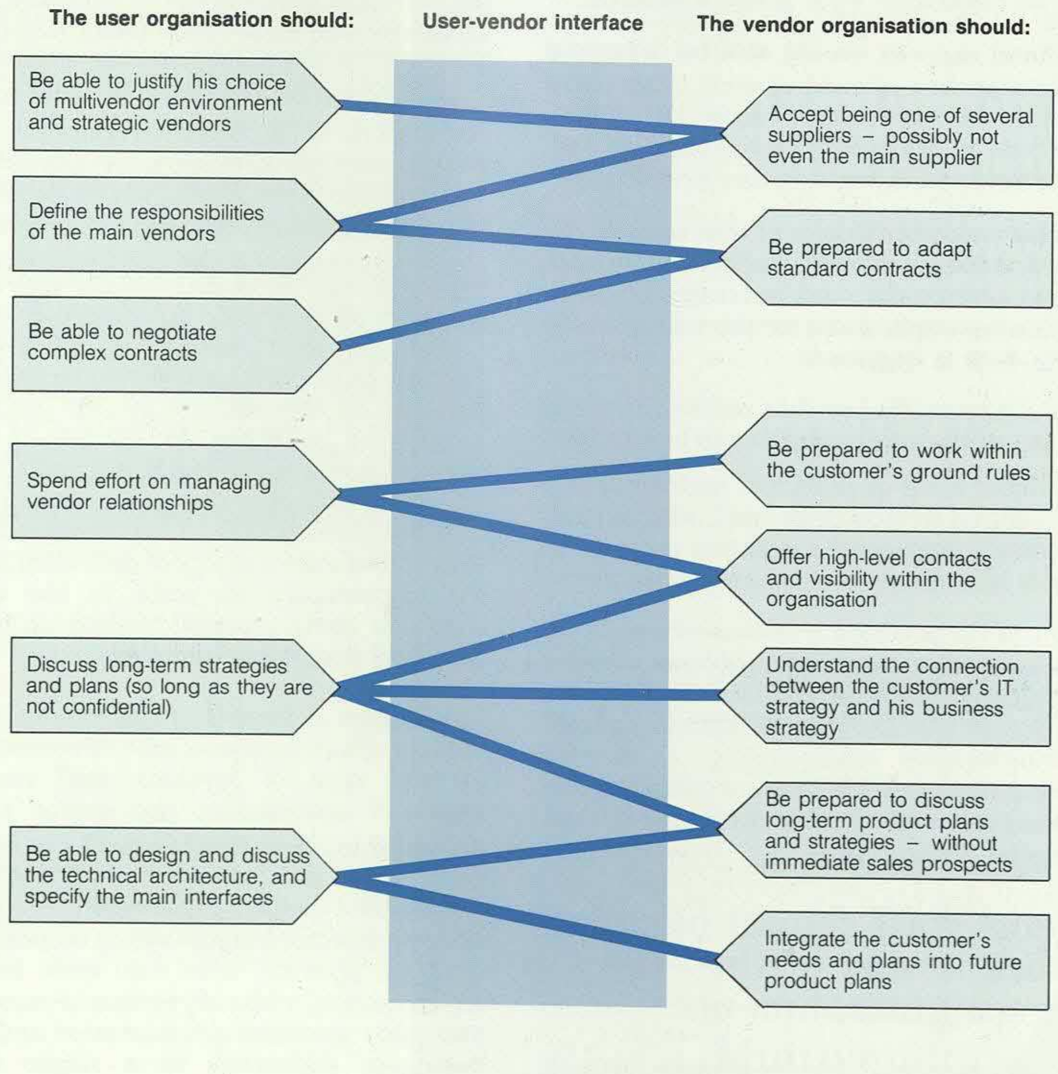
Type of integration	Critical elements of the techical architecture	Means of implementation: Batch interchange	Means of implementation: Realtime
Data-driven	Data management and data structures Communications software	Duplicate data	File-server architecture Database machine Distributed database
User-driven	Network architecture Communications software User-interface standards and software	Not appropriate	Remote-session access Protocol conversion Common user interface
Process-driven	Applications architecture Communications software	File-transfer	Terminal emulation Client-server architecture Remote procedure call Cooperative processing

systems development skills. They involve an understanding of the way the business might develop, of the role that IT can play, and of the technology itself. Either systems staff will need to be trained in this specific area, or external assistance will need to be sought, perhaps from consultants or from systems integrators.

The technical architecture should not be defined by systems staff alone, however. The involvement of *users* will ensure that the technical architecture can evolve to accommodate likely new business requirements, and that the users understand that they may need to restrict their choices in order to be able to integrate their applications with those of other business units. The involvement of *vendors* will ensure that the systems function knows about current and planned products and gets sound advice on the most appropriate products from a vendor's range.

The relationship with vendors should be a partnership. This should be encouraged by a series of regular and ad hoc meetings between the systems function and the vendors, ranging from operational meetings to resolve day-to-day problems, to strategy meetings involving senior managers or even board members in the two organisations. Figure 6 illustrates the nature of the dialogue that should occur at these meetings. While vendors will usually be willing to help resolve any difficulties, the systems function is ultimately responsible for making sure that the products from the different vendors can work together.

Figure 6 A working partnership in a multivendor environment requires a dialogue between user and vendor organisations



Users should not be aware of the different software environments

From the users' perspective, a basic requirement in a multivendor environment is for one terminal that can be used to access all the applications and data, regardless of the software environment in which a particular application or set of data is physically located. It is also important that the user interface is the same for applications running in different software environments. The need for consistency is difficult to achieve in a multivendor environment, and will usually require quite extensive own coding. We expect that most of the major computer suppliers will adopt broadly similar user interfaces, although there will be three or four main standards with which different groups of suppliers will comply. In-house developments, as well as bought-in software, should be consistent with these standards.

Operational support should also be organised with the objective of making users unaware of the various software environments. To this end, users should be provided with a single help desk and standardised administrative procedures.

In a multivendor environment, the work of the help desk is complicated by the need to provide a unified service for each software environment. This has implications for the way in which the help desk is organised:

- It must be staffed so that problems arising in any of the environments can be handled.
- A higher level of skills and understanding is required by help desk staff, who have to be able to interpret a problem in order to know who should be contacted to resolve it.

Procedures for getting a terminal connected to the computer, for reporting problems, and for producing accounting information should, as far as possible, be identical for the various systems in a multivendor environment. A lack of standard procedures will be more confusing and frustrating in a multivendor environment than in a simpler environment.

Systems staff should be able to work in any of the software environments

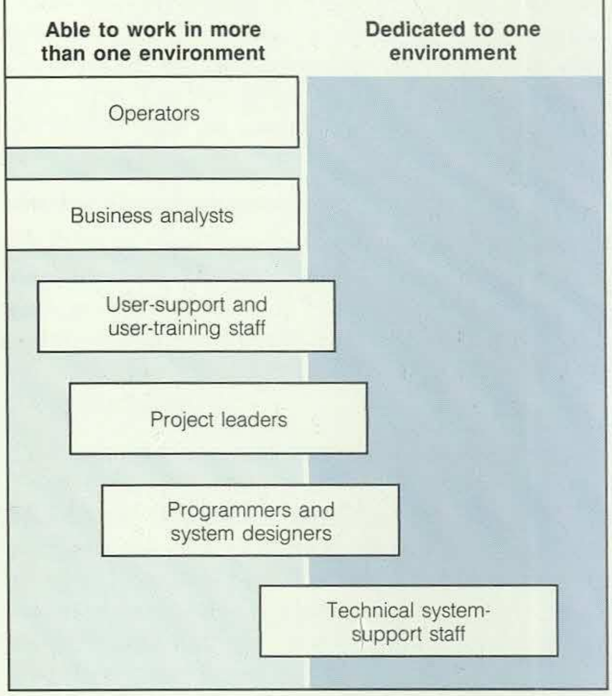
Staffing the systems function is more difficult in a multivendor environment because different technical skills are required for each software environment. Ideally, systems staff should have a range of skills that enable them to work in any one of the software environments. This will give managers much greater flexibility to allocate staff to projects according to the needs of the business, and at the same time, is likely to reduce staff turnover and increase motivation.

It is not, of course, easy to build a team of systems staff that can be used in such a flexible way, especially in view of the fact that it will become more and more difficult to recruit qualified staff. Instead of limiting the sources of recruits to those with specific technical skills, staff with the *potential* to perform the broader roles required in a multivendor environment should be recruited, and then trained to provide them with the necessary capabilities and skills. Doing this will result in better utilisation of staff (and hence a more cost-effective service to users), increased sources of potential recruits, and improved staff competence. There will be barriers to be overcome in adopting this approach, but in view of the fact that continuing staff costs are by far the biggest cost item in a multivendor environment, it is essential that it be implemented if costs are not to increase to an unacceptable level.

Selecting candidates with the required potential requires appropriate recruitment procedures, such as the use of personality measurements, job and personnel specifications, and regular reviews of the success of the recruitment process by following the careers of recruits. Once a promising candidate has been recruited, his or her potential should be developed to fit into the staffing plan of the systems function. This will require a policy of providing staff with the opportunity to work in the different software environments, backed up by a well designed training programme.

The degree of flexibility that can be achieved through these measures will, of course, depend on the type of systems staff concerned. Figure 7 summarises the extent to which different types of systems staff can be used in more than one software environment. We recommend that in a multivendor environment, systems directors should aim to employ as many staff as possible who can work in several environments, with only technical-support staff, and some specialist programmers and systems designers dedicated to a single software environment.

Figure 7 Most types of staff can be assigned to work in more than one software environment



Striking a balance between the business pressures to increase the number of vendors and the extra costs and technical complexities that will be incurred in integrating applications in the resulting different software environments is not an easy task. Being aware of the cost and technical implications will enable systems directors to make better-informed decisions about moving to, or staying in, a multivendor environment, and to ensure that the business benefits arising from a multivendor environment are obtained without incurring an unnecessarily high cost penalty.

Butler Cox

Butler Cox is an independent management consultancy and research organisation, specialising in the application of information technology within commerce, government, and industry. The company offers a wide range of services both to suppliers and users of this technology.

Butler Cox Foundation

The Butler Cox Foundation is one of the services provided by Butler Cox. It provides the executives responsible for information systems in large organisations with a continuous analysis of major developments in the technology and its application.

The Foundation publishes six Research Reports each year together with a series of special Directors' Briefings and Position Papers. The programme of activities includes a wide range of meetings that provide Foundation members with a regular opportunity to exchange experiences and views with their counterparts in other large organisations.

Recent publications

Research Reports published recently are:
Using System Development Methods
Senior Management IT Education
Electronic Data Interchange
Expert Systems in Business
Competitive-Edge Applications: Myths and Reality
Communications Infrastructure for Buildings
The Future of the Personal Workstation
Managing the Evolution of Corporate Databases
Network Management
Marketing the Systems Department
Computer-Aided Software Engineering (CASE)
Mobile Communications
Software Strategy
Electronic Document Management
Staffing the Systems Function

Recent *Directors' Briefings* are:
Information Technology and Realpolitik
The Changing Information Industry
Using Technology to Improve Competitive Ability
1992: An Avoidable Crisis
Managing Information Systems in a Decentralised Business

Recent *Position Papers* are:
Document Image Processing — The Next Revolution?
A Progress Report on New Technologies
Hypertext
Pan-European Communications: Threats and Opportunities

Additional copies of publications

Members of the Butler Cox Foundation receive several copies of each publication, including the Management Summaries for each Research Report. Additional copies of all publications may be purchased by members from Butler Cox.

BUTLER COX FOUNDATION

Butler Cox plc
Butler Cox House, 12 Bloomsbury Square,
London WC1A 2LL, England
☎ (01) 831 0101, Telex 8813717 BUTCOX G
Fax (01) 831 6250

Belgium and the Netherlands
Butler Cox BV
Burg Hogguerstraat 791,
1064 EB Amsterdam, The Netherlands
☎ (020) 139955, Fax (020) 131157

France
Butler Cox SARL
Tour Akzo, 164 Rue Ambroise Croizat,
93204 St Denis-Cédex 1, France
☎ (1) 48.20.61.64, Télécopieur (1) 48.20.72.58

Germany (FR)
Butler Cox GmbH
Richard-Wagner-Str. 13, 8000 München 2, West Germany
☎ (089) 5 23 40 01, Fax (089) 5 23 35 15

United States of America
Butler Cox Inc.
150 East 58th Street, New York, NY 10155, USA
☎ (212) 891 8188

Australia and New Zealand
Mr J Cooper
Butler Cox Foundation
Level 10, 70 Pitt Street, Sydney, NSW 2000, Australia
☎ (02) 223 6922, Fax (02) 223 6997

Finland
TT-Innovation Oy
Meritullinkatu 33, SF-00170 Helsinki, Finland
☎ (90) 135 1533, Fax (90) 135 1091

Ireland
SD Consulting
72 Merrion Square, Dublin 2, Ireland
☎ (01) 766088/762501, Telex 31077 EI,
Fax (01) 767945

Italy
RSO Futura Srl
Via Leopardi 1, 20123 Milano, Italy
☎ (02) 720 00 583, Fax (02) 806 800

Scandinavia
Butler Cox Foundation Scandinavia AB
Jungfrudansen 21, Box 4040, 171 04 Solna, Sweden
☎ (08) 730 03 00, Fax (08) 730 15 67