Management Summary

BUTLER COX FOUNDATION

Software Strategy

BUTLER COX
FOUNDATION

Software Strategy

Management Summary
Report 69, May 1989

Butler Cox & Partners Limited

LONDON
AMSTERDAM MUNICH NEW YORK PARIS

Published by Butler Cox & Partners Limited
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.

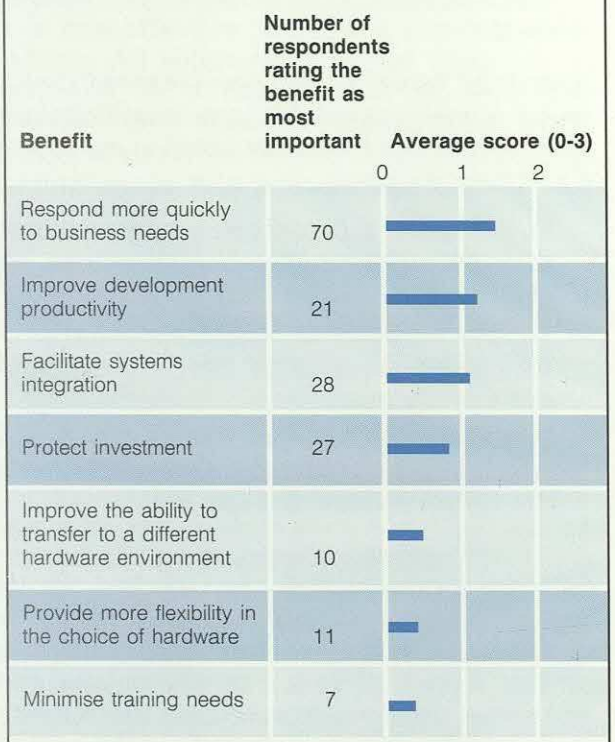
Management Summary

This document summarises the main management messages from Foundation Report 69, published in May 1989. The full report is available to members of the Butler Cox Foundation.

Software is an increasingly important investment for most organisations, but one that is inherently risky and difficult to control. Choosing the right strategy for investing in new software and managing the existing software base are therefore important management concerns.

Like any strategy, a software strategy must define the desired goals. Too many systems departments, however, still define their software goals in technical terms (to convert all applications from IMS to DB2, for example, or to use fourth-generation languages for all new applications). Instead, the software goals should be defined in terms that help other parts of the business achieve the organisation's *business* goals. Examples would be to allow all staff in the finance department to have instant access to important data, or to enable customers to place orders electronically. Specifying software-strategy goals in terms such as these ensures that IT is perceived as making a real contribution to the business, and that the software strategy is an integral part of the business strategy. Unless software goals are expressed in business terms, it will not be possible to obtain the full benefits from a software strategy.

Figure 1 **The ability to respond more quickly to business needs is seen as the most important reason for having a software strategy**



Respondents were asked to rate the importance of each benefit on a scale of 0 to 3

(Source: Survey of 170 Foundation members)

A software strategy provides significant benefits

The benefits that Foundation members expect to gain from having a software strategy are shown in Figure 1. These correspond largely with the five main benefits identified by our research:

— It permits the systems department to respond faster to demands from the business for new or enhanced systems. Because the strategy will reduce the variety of software being used, the systems department can concentrate its skills on a smaller number of technical areas and leave development staff free to work on business applications, rather than on special programs to interlink different types of software.

— It encourages an organisation to manage its software as a business investment. Software, like other business assets, has a finite life and has to be replaced as it ages. Expenditure on software therefore continues to be a large, and increasing, part of the overall IT budget. A software strategy helps to minimise this expenditure, by ensuring that unnecessary products are not purchased, by minimising the costs of replacing (or renewing) applications, by allowing bulk discounts to be

obtained, and by reducing training costs both for systems staff and for users.

- A software strategy allows different software applications to interwork. For many organisations, lack of interworking is a major problem that prevents them from making the best of their software products.
- It allows a more flexible choice of hardware, because equipment from different suppliers can be used to run the same applications.
- A clearly defined software strategy helps to retain skilled and difficult-to-replace systems staff.

A strategy, however, can provide benefits only once it is implemented. We have identified three main actions that have to be taken to implement a software strategy:

- To define the software components required to develop and run specific applications. We call this the *software infrastructure*.
- To select the set of *software standards* to which the elements of the infrastructure must conform.
- To set the *software-procurement policy* to ensure that new software is compatible with the infrastructure and the standards.

A software strategy, based on an appropriate software infrastructure, and backed up by professional standards for constructing new applications, will allow user departments to construct more of their own applications and will ensure that they do not repeat the mistakes made by the systems community in the past 20 or 30 years.

The software infrastructure has five main components

Developments in software products are making it possible to create a software infrastructure and to use it as the basis for a software strategy. Originally, there was a clear distinction between system-software and application-software products, and most of the functions required by an application had to be specifically coded. Today, however, the boundary is less clear — an application package, for example, may be based on a well-known database management system. This means that development staff can now use infrastructure software as the basis for many of the functions that previously had to be specifically coded (see Figure 2). In turn, this means that, increasingly, applications will be constructed (rather than developed) by assembling the appropriate infrastructure components and adding the bespoke functions to meet the users' business requirements.

The software infrastructure has five main components. These are illustrated in Figure 3 and are discussed below.

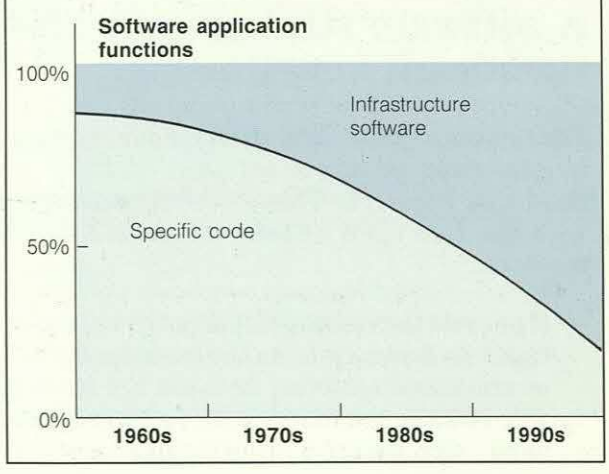
The *development and operating environment* includes the operating system, machine utilities, system development tools, performance tools, and any other facilities needed to develop, implement, and run software efficiently. Choosing the right products for this part of the infrastructure is possibly the most difficult decision the systems director has to make on software strategy.

The *data-management* component comprises a database management system and data-access tools. Increasingly, these will be based on relational systems and SQL, the *de facto* standard for accessing relational databases.

The *communications* component includes software functions for both internal and external communications. Often, these are based on proprietary network architectures such as IBM's SNA and DEC's DECnet.

The *user-interface* component comprises software and in-house standards and guidelines that ensure that, to the users, all applications look and feel the same. At present, there are few software products available for the user-interface component of the infrastructure, although IBM's Presentation Manager will be a strong

Figure 2 **An increasing proportion of the software functions required by an application is provided by the software infrastructure rather than by specific code**



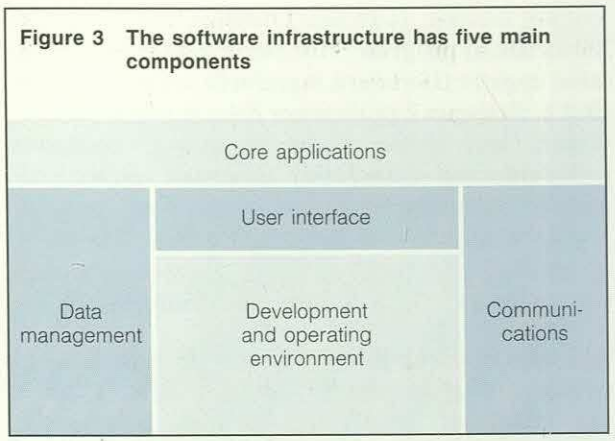


Figure 3 The software infrastructure has five main components

contender for establishing *de facto* standards for personal computers and workstations.

Core applications are essential to the day-to-day operation of the business. Usually, they store or modify data held in corporate databases. An application that accesses corporate data, but that creates data only for its own use, will usually not be a core application, and therefore not part of the software infrastructure. Core applications are managed by the systems department as part of the software infrastructure. Non-core applications will increasingly be constructed by the user community, using the software infrastructure for this purpose. This does not imply that core applications are more important than non-core applications. Indeed, competitive advantage is often derived from non-core applications.

The greatest benefits will, of course, be obtained if the number of infrastructures is kept to a minimum. One is the ideal, although this can rarely be achieved in practice, because different infrastructures may be required for specialised applications (scientific computing, or office systems, for example). We recommend that additional infrastructures should be implemented only when at least one of the following conditions applies:

- There is no requirement to link applications in the different infrastructures.
- The additional infrastructure, and its applications, can be managed by the users or by a third party.
- Communications between the infrastructures can be handled by simple file-transfer and conversion facilities.

Most Foundation members will need to migrate to a new software infrastructure over a period of time. The most cost-effective way of doing this is to take a series of small steps based on the infrastructure needs of new applications. The infrastructure components can therefore be acquired to meet the requirements of new applications as they are developed, with the development priorities being set according to the level of business benefits provided by the applications.

However, as the components of the infrastructure are progressively implemented, it may be possible to bring forward the point at which it is cost-effective to replace existing applications. All applications should therefore be reviewed on a regular basis to determine whether they should continue being maintained, or should be replaced using (and perhaps adding to) the new infrastructure.

Software standards ensure that the infrastructure components are compatible

The second main action in implementing a software strategy is to select the set of software standards that will ensure that the infrastructure components are compatible and can interwork. Significant progress is being made in both open and proprietary standards. Open standards, particularly those based on Unix, are an increasingly viable choice, especially for intelligent workstations and minicomputers. For mainframes, the most significant recent development is IBM's Systems Applications Architecture (SAA).

A family of standards is required, covering all the components of the software infrastructure. However, setting software standards has some disadvantages as well. The more rigid the standards, the smaller the choice of software products that comply with the standards. Moreover, products complying with the standards may have fewer functions or be less easy to use than non-standard products. In addition, products complying with an all-embracing standard will be complex, expensive, and less efficient than those that are optimised for a narrow range of functions.

Another problem with standards is the slow progress being made with public open standards. It can take four years or longer to gain full approval for a new ISO standard. Suppliers are not prepared to wait that long to bring new

technology to the market, and it is inevitable that new-technology products will be based on proprietary standards.

Unix-based standards are becoming viable, but not yet for mainframes

Unix systems are now well-established in the networked-workstation environment and are increasingly being used in 'niche' application areas. The price-performance of Unix systems is often significantly better than for systems based on proprietary operating systems, because of the highly competitive nature of the Unix market. Three organisations have been formed by various groups of suppliers to promote Unix and other open standards. Two of these are supporting different versions of Unix. Unix International promotes AT&T's version, while the Open Software Foundation is dominated by other major hardware vendors, including IBM and DEC, who are promoting a version based on IBM's AIX. Both versions, however, are compatible with the portability guide published by X/Open, the third organisation promoting Unix, but they offer different extensions and additional facilities. It is too early to say if the work of any of these groups will lead to a *de facto* (or even a *de jure*) Unix standard.

Despite these developments, Unix is not likely to supersede established mainframe operating systems like IBM's MVS and DEC's VMS in the short to medium term. Although, in theory, Unix could soon be a viable alternative, in practice, major user organisations will not be prepared to contemplate a migration to Unix. Their existing investments in hardware, software, and skills are so large that the cost, time, and effort involved in moving away from their existing mainframe environments rules out the possibility.

In addition, major suppliers who provide both Unix and proprietary environments go to great lengths to emphasise the superiority of their proprietary products for mainstream corporate computing. Thus, IBM and DEC have restated their commitment to MVS and VMS respectively, and are particularly concerned to position Unix as an unsuitable base for large-scale commercial applications.

IBM's SAA is an important initiative

In most countries where there are Foundation members, IBM is the dominant mainframe supplier. Even in France, where Bull is the market leader, it is not possible to ignore the importance of IBM's *de facto* standards. The most important recent standards initiative from IBM is Systems Application Architecture (SAA), which has three main elements: common programming interface, common communications support, and common user access. Figure 4 shows how they interrelate. The SAA concept also includes common applications able to run in any SAA hardware environment.

It is important to realise, however, that SAA is not a set of product specifications; it is a collection of selected software interfaces, conventions, and protocols. The full benefits of SAA will be obtained only when there is a full range of supporting software products, and it will be several years before this is achieved. User organisations are therefore unlikely to be able to commit to SAA until 1992 at the earliest.

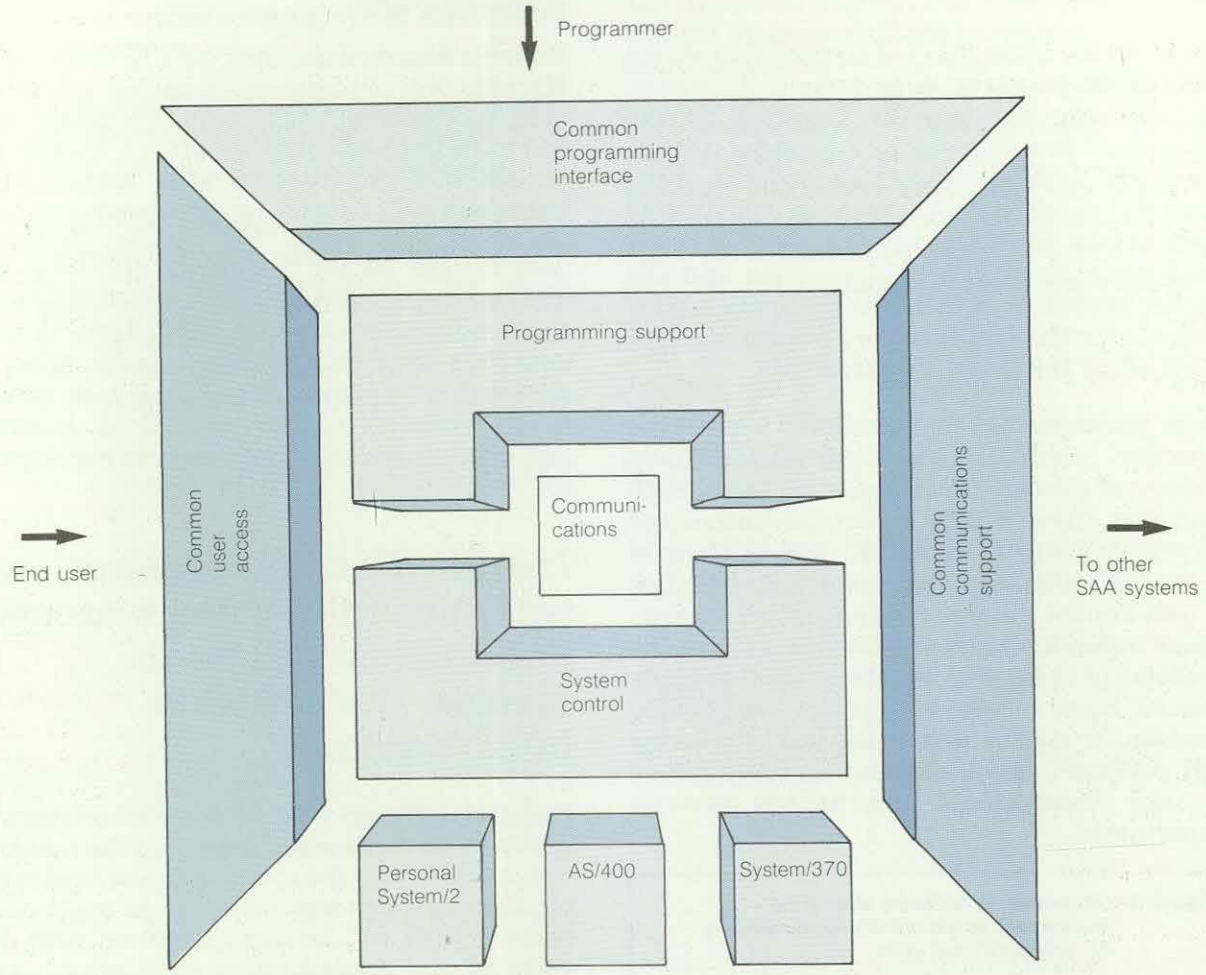
Even if a full range of products were available today, many organisations would not be able to move quickly to a new SAA environment. Several commonly used IBM hardware ranges and standards are excluded from SAA. Existing investments in these 'SAA orphans' will mean that user organisations will need to migrate to SAA over several years. Nor will it be possible to standardise on SAA *per se*. The scope of SAA is broad and is growing. Software suppliers and user organisations will therefore need to select a subset of SAA, and there will be considerable scope for incompatibilities between products in which different subsets are implemented.

In the short term, the main benefit of SAA will be derived from the common user access standards. This element of SAA is designed to make applications have a similar look and feel, regardless of the hardware on which they are running. IBM's Presentation Manager for the OS/2 operating system, for example, supports SAA's common user access.

Product-based standards are to be preferred

The all-embracing nature of open standards, specified at great length and in great detail by standards-making bodies, means that user organisations must select a subset of the facilities. The same applies to an all-embracing proprietary architecture like SAA. Another drawback of proprietary standards is that the vendor controls them and may change them at will. Standards based on successful products,

Figure 4 SAA is not a set of products; it is a collection of conventions, interfaces, and protocols



(Source: IBM)

especially *de facto* standards, tend not to suffer from these problems to the same extent.

For most practical purposes, the most effective type of standards are those encapsulated in software-infrastructure products. Such standards are clearly defined by the way the product behaves, and it is much easier to test if new software conforms to the standard: either it can interwork with the 'standard' product, or it cannot.

A software-procurement policy ensures new software matches the infrastructure

The third main action in implementing a software strategy is to set the software-procurement policy. This will specify the options for procuring new software to ensure that it matches the software infrastructure and standards. A prerequisite for such a policy is to decide on the most appropriate organisational level for making various types of decision about software. It is necessary to strike the right balance between centralised specification of standards and decentralised decisions about software procurement. Figure 5, overleaf, shows which types of decisions can be centralised and decentralised in four types of organisation.

There are several ways in which software can be procured, but for most organisations, the main choice for applications software will be to use a package or to develop a bespoke system. We believe that much greater use can be made of packages than has traditionally been the case.

Packages are often a better investment than bespoke development

Packages are often rejected because they do not meet all of the users' requirements. However, by assessing the benefits provided by the *additional* functionality of a bespoke system, compared with a package that meets all of the *essential* requirements, a package will often be seen to be a far better investment. The lower implementation costs of a package and the earlier provision of benefits more than compensate for the fact that the package does not meet all of the users' requirements.

'Soft' packages that can be tailored to meet the specific needs of an organisation have weakened the case for not using packages. Such packages often provide report-generation and screen-formatting facilities, and a fourth-generation language. They are usually based on a well-known database management system, which makes it easier to extend the scope of the package, or to develop interfaces to other applications. Some software suppliers now provide workbench tools to assist in the task of tailoring soft packages. Some of these tools are aimed at user departments, not at the systems department.

Figure 5 Decisions on software should be centralised or decentralised, depending on organisational style

Organisational style	Standards and policies	Software infrastructure (including core applications)	Software procurement	Non-core applications
Centralised	C	C	C	C
Coordinating	C	C	C	D
Guiding	C	C	D	D
Devolved	C	D	D	D

C = centralised
D = decentralised

User departments should be encouraged to construct applications

Systems departments must not ignore the trend towards user departments wanting, and being able, to construct their own applications, and software-infrastructure products should be chosen with this trend in mind. Resisting the trend will only encourage user departments to 'do their own thing', but without adequate professional advice, standards, and controls. The systems department should therefore help users to choose appropriate packages and to construct their own non-core applications. It should also set standards and guidelines for the user community, and provide a quality-assurance function to help users to conform to the standards.

In time, the main applications-software role of the systems department will be to manage the software infrastructure

Basing the strategy on a software infrastructure will have a fundamental impact on the role and responsibilities of the systems department with respect to applications software. In particular, systems staff will be less concerned with developing and implementing applications, and more concerned with defining and managing the software infrastructure.

The need for technical skills will not disappear, however. Choosing and managing a comprehensive infrastructure is a difficult task that requires highly technical skills in the areas of databases, network management, software engineering, and so forth. Even when most applications are constructed by user departments, programmers will still be required to develop interface and conversion software. The skills needed in future will principally be those of systems integration. In summary, the main responsibilities of the systems department will be:

— To devise and manage the procedures for migrating to the new software infrastructure and for converting existing applications to conform to it.

— To ensure that software-infrastructure components provide adequate capacity, performance, reliability, and availability.

- To set standards for using the software infrastructure to build new applications or enhance existing ones, and to provide high-level consultancy support for application constructors.
- To resolve difficulties that might arise from allowing user departments to make their own software-procurement decisions.
- To devise a chargeback mechanism that encourages the user community to use the software infrastructure in the most effective way.
- To plan the evolution of the software infrastructure.

Systems departments must accept as a fact of life that business staff will increasingly have the skills, and access to the tools, to enable them to construct more of their own applications. They should therefore start planning for, and encouraging user involvement in, software selection and construction. The first step is to ensure that the organisation has an appropriate software infrastructure that is backed up by professional standards for using it to construct new applications.

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 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

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 *Technology Briefings* are:
Document Image Processing — The Next Revolution?
A Progress Report on New Technologies
Hypertext

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 & Partners Limited
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
3rd Floor, 275 George Street, Sydney 2000, Australia
☎ (02) 236 6161, Fax (02) 236 6199

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

The Nordic Region
Statskonsult AB
Stora Varvsgatan 1, 21120 Malmo, Sweden
☎ (040) 1030 40, Telex 12754 SINTABS

Spain
Associated Management Consultants Spain SA
Rosalía de Castro, 84-2°D, 28035 Madrid, Spain
☎ (91) 723 0995