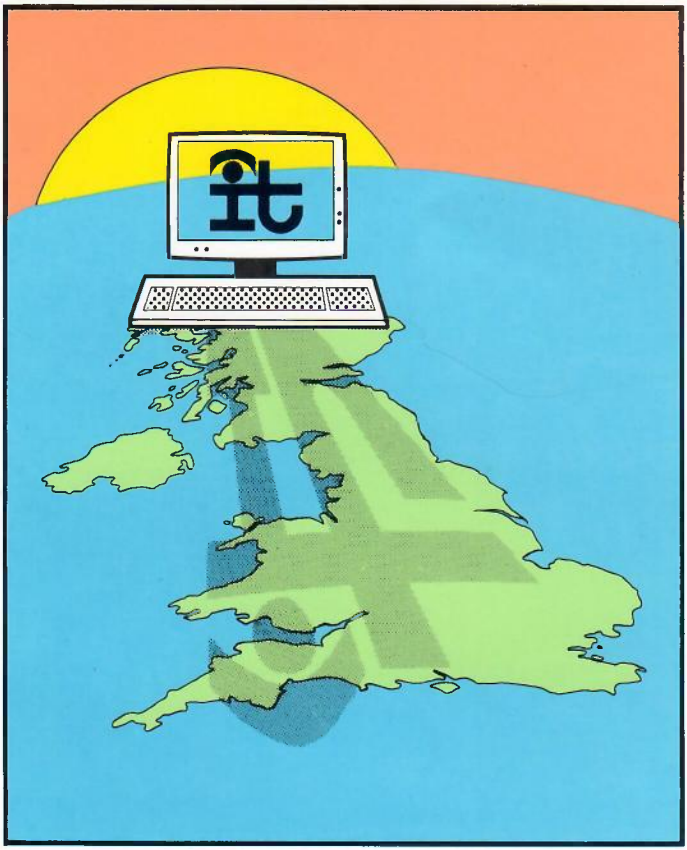
An ACARD report

SOFTWARE

A vital key to UK competitiveness



Cabinet Office: Advisory Council for Applied Research and Development

The Advisory Council for Applied Research and Development (ACARD) has the following terms of reference:

'To advise the Government and publish reports as necessary on:

i. applied research, design and development in the United Kingdom;

ii. the application of research and technology, developed in the United Kingdom and elsewhere, for the benefit of both the public and private sectors in accordance with national economic needs;

iii. the co-ordination, in collaboration with the Advisory Board for Research Councils, of these activities, with research supported through the Department of Education and Science;

iv. the role of the United Kingdom in international collaboration in the fields of applied research, design and development related to technology.'

The members of the Council are:

Sir Francis Tombs F Eng (Chairman)	Chairman, Rolls Royce Ltd.
Mr M. Bullock	Corporate Finance Director, Barclays Bank PLC.
Mr J F Coplin F Eng	Director of Design, Rolls Royce Ltd.
Sir Kenneth Durham	Chairman, Unilever PLC.
Mr J W Fairclough	Director of Manufacturing and Development IBM (UK) Ltd, Chairman IBM (UK) Laboratories.
Mr A K Gill	Group Managing Director, Lucas Industries PLC.
Mr E Hammond	General Secretary, Electrical, Electronic, Telecommunications, and Plumbing Union.
Dr B C Lindley CBE	Director of Technology, BICC Ltd.
Mr G J Lomer CBE	Technical Director, Racal Electronics PLC.
Mr R Malpas CBE F Eng	Managing Director, British Petroleum Company PLC.
Professor J S Metcalfe	Professor of Economics, Manchester University.
Professor Sir David Phillips FRS	Professor of Molecular Biophysics, University of Oxford; Chairman of the Advisory Board for the Research Councils.
Dr C H Reece	Director of Research and Technology, Imperial Chemical Industries PLC.
Sir Rex Richards FRS	Director, Leverhulme Trust.
Professor S D Smith FRS FRSE	Professor of Physics, Heriot Watt University; Chairman, Edinburgh Instruments Ltd.

Cabinet Office
Advisory Council for Applied Research and Development

Software: A vital key to UK competitiveness

London Her Majesty's Stationery Office

First published 1986

ISBN 0 11 630829 X

Contents

Executive summary

E.1 World-wide, the manufacturing and service industries are increasing their competitiveness by accelerating their use of computer software in end products, manufacturing processes and customer services. United Kingdom manufacturing and service industries can regain and enhance their competitiveness by applying software more widely. To apply software effectively, the UK industries require a better understanding of the nature and power of software; this understanding can only be achieved through increased in-service education and training. More knowledgeable manufacturing and service industries will be more able to pull through and take up new ideas from the supply side, which itself needs strengthening educationally.

E.2 The world software market is large ($40 billion) and growing rapidly (30% pa). The UK market is 5% of the world market. UK firms hold only 2-3% of the world market. In a period of rapid growth, the challenge for the UK software industry is to build a firmer foundation for other UK industries, and win a greater market share so as to reverse the emerging software balance of payments deficit.

E.3 Our recommendations are designed to help UK industry profit from the opportunities which software is creating, without recourse to government intervention. Our constructive approach has generated both strategic targets and detailed, tactical recommendations. Companies, government departments and educational establishments need clear visibility of the broad picture if they are to develop a co-ordinated approach to achieving the overall targets.

E.4 We recommend the formation of an expert body, provisionally called STARTING (Software Technology and Applications Review Team of INdustry and Government), to monitor the implementation of our recommendations and their effectiveness in achieving the targets we have set. STARTING's main function is to hold an annual, large scale, formal review meeting to consider a performance report of software users, appliers and suppliers. ACARD requires an annual summary report from STARTING in order to monitor its targets.

E.5 Our main recommendations for industry are:

- ⋆ In-service training initiative for all users, appliers and suppliers;
- ⋆ Increased application of software to increase the competitiveness of the manufacturing and service industries;
- ⋆ Marketing initiative for UK software products and services.

E.6 Our main recommendation for government is for inter-departmental co-operation on:

- ⋆ Public purchasing to exercise demand-side leadership;
- ⋆ Technology transfer acceleration;
- ⋆ Better R&D planning to avoid discontinuities in policy and funding;
- ⋆ In-service training initiative for government employees.

Foreword

The United Kingdom's manufacturing industries, service industries, defence, public services and domestic life are critically dependent on Information Technology (IT). IT is the construction and operation of information handling systems composed of microelectronics, computers, software and telecommunications. Of these constituents, software is the most complex, most costly and most difficult component to develop and bring to market. Software Engineering is the name given to the methods, skills and tools used to develop software.

The world market for IT products is large and growing rapidly. Major industrial nations are vigorously developing, promoting, and exploiting IT. Software is a key strategic element in the fierce economic competition between such nations.

The attention of ACARD was drawn to the possibility that the UK was failing to develop and apply software sufficiently rapidly and widely in all industries. A related possibility was that the UK was failing to win its expected share of the world market for software.

An ACARD Working Group with the following composition was therefore set up in January 1985 to investigate these concerns:

Mr J Coplin (Chairman)	Director of Design, Rolls Royce Ltd.
Miss S Bond	Royal Signals and Radar Establishment, Ministry of Defence.
Mr D Collens	Technical Director, Software Sciences Ltd.
Mrs J Connell	General Manager, F International Ltd.
Prof C A R Hoare	Programming Research Group, Oxford University.
Mr G W Holmes	Deputy Chairman, Systems Designers PLC.
Lt-Col P Ost	Procurement Executive, Ministry of Defence.
Mr D Robson	Controller, Software Support Services, National Westminster Bank PLC.
Dr M F Smith	Head of R&D, ISTEL Ltd.
Mr J Whalley	Technical Manager, British Aerospace, Manchester.
Dr C Whitby-Strevens	Microcomputer Support Manager, Inmos Ltd.
Dr R W Witty	Deputy Director (Software Engineering), Alvey Programme.

The terms of reference of the Working Group were:

i. to identify measures to assist in the understanding and use of software engineering by industry and business through education and training;

ii. to report on the nature of existing and prospective management and organisation practices in software development, in the light of the changes being and likely to be brought about by software engineering;

iii. to report on current and prospective techniques and aids in the specification, design, integration and testing of software in the assessment of quality and correctness of product;

iv. to report on the international competitive position of the United Kingdom with respect to the above, including restrictions on international trade in software and its implications for UK industry and business;

v. to make recommendations.

Although the remit provided for the Working Group set down specific objectives for the study, it was considered necessary, in the event, to adopt a broader interpretation. The approach taken was to seek to identify the macro-economic significance of software to the United Kingdom; then to develop an understanding of the actions required to produce a better market performance; and how to develop the methods, skills and tools to achieve better software technology. Particular emphasis was placed on the users of software and on the collective behaviour of users, appliers and suppliers of software. This provided a framework within which the necessary steps to bring about the widespread use of software, and software engineering, could be achieved.

The Working Group's report was considered in February 1986 by the Council, who endorsed its conclusions and recommendations. It will be published, with the approval of Government, to draw attention to the importance of software, to the concerns arising from the findings and to facilitate wider discussion of the Working Group's recommendations.

The Council is grateful to the members of the Working Group for their contribution to ACARD's work, to the many individuals and companies who gave evidence to the Working Group, and the support of the ACARD Secretariat in the Cabinet Office.

March 1986

1. Summary of conclusions and recommendations

The challenge to the UK's competitive position

1.1 A vital opportunity exists for the manufacturing and service industries to improve their competitiveness by increasing their use of computer software in end products, manufacturing processes and customer services. A better understanding of the power of software is needed before it can be successfully applied more widely and effectively. Not enough understanding exists, at all management levels, of the opportunities for exploiting software in products and processes. Nor is there enough understanding of modern techniques for software application and development. These deficiencies are mainly due to a lack of ongoing, in-service education and training, at all levels of management and technical staff.

1.2 The software supply industry operates in a world market worth some $40 billion per annum, with a soaring growth rate of around 30% per annum. The UK domestic market is 5% of the world total. UK suppliers hold 2-3% of the world market. This share consists of about half of the UK domestic market plus only 0.5% of the rest of the world market. Many new UK software companies have been created in the last twenty years. Several have grown rapidly, at around 20% per annum, which compares very favourably with other UK industrial sectors. In spite of this relative success, the major UK software companies are much smaller than their major overseas competitors. Thus, in a period of rapid world market growth, the challenge facing the UK software industry is how to win a greater market share and build a firmer foundation from which to survive and prosper during the 'shake out' phase of market development which is now dawning.

1.3 In framing our conclusions and recommendations we are conscious of the following concerns:

* The UK manufacturing and service industries (as well as defence, utilities and public administration) are now dependent on Information Technology (IT).
* Almost every aspect of business in the UK is open to international competition.
* In the economic struggle between America, Japan, the Far East and Europe, the use of IT is a major weapon.

- ⋆ IT is critically dependent on software.
- ⋆ The UK software market is dominated by foreign suppliers. This has three consequences:
  a. The UK can no longer be sure of obtaining the software its manufacturing and service industries need to sustain their international competitiveness.
  b. The increasing demand for imported software will bring about a major new source of balance of payments deficit.
  c. The dominance of the UK and world markets by overseas suppliers is weakening the UK software industry. The UK industry is not effectively fighting back.
- ⋆ Should the UK be unable to afford, or not be allowed, to import foreign software needed to remain competitive, then the UK software industry will not be able to deliver competitive alternatives and so the whole of industry will be weakened.

1.4 While much ground has already been lost, prompt action can recover and improve the situation. If our recommendations are vigorously implemented, we believe no recourse to government intervention or subsidy will be necessary.

STARTING – A framework for all recommendations

1.5 The widespread application of soundly engineered software can enhance the overall financial performance of the United Kingdom. To maximise this potential, companies, government departments and educational establishments need clear visibility of the broad picture. Thus informed about national targets, and the general direction to be taken, they can then strive to achieve the overall targets by mutually co-ordinated actions.

1.6 **We recommend** the formation of an expert body, provisionally called STARTING (Software Technology and Applications Review Team of INdustry and Government), to monitor the implementation of our recommendations and their effectiveness in use against the targets we have set. STARTING's main function is to hold an annual, large scale, formal review meeting to consider a performance report of software users, appliers and suppliers, ACARD requires an annual summary report from STARTING in order to monitor the effectiveness of the STARTING mechanism and the progress towards its targets.

Main recommendations

1.7 Our main recommendations for industry are:

- ⋆ in-service training initiative for all users, appliers and suppliers;

* increased application of software to improve the competitiveness of the manufacturing and service industries;
* marketing initiative for UK software products and services.

1.8 Our main recommendation for government is the formulation and implementation of a long-term plan (TYPSSEA) for interdepartmental co-operation on:

* public purchasing to exercise demand-side leadership;
* a new technology transfer initiative (BASSMATT);
* better R&D planning;
* in-service training initiative.

1.9 The widespread application of soundly engineered software has the potential to enhance the national competitiveness of the UK. In Chapter 7 we make our main recommendations which we believe will maximise this potential. These main recommendations are divided between industry and government. In the three subsequent chapters we give more details about these main recommendations and make further recommendations. Chapter 8 contains the detailed recommendations for all industries and for the user, applier and supplier roles specifically. Chapter 9 contains the detailed recommendations for government; these are assigned to individual government departments. Chapter 10 contains detailed recommendations for the professional institutions. A 'routemap' of the organisation of our recommendations is given in Figure 1. The Appendices contain detailed discussion of three topics: in-service education and professional qualification; the software implications for safety and standards; and product standards and certification.

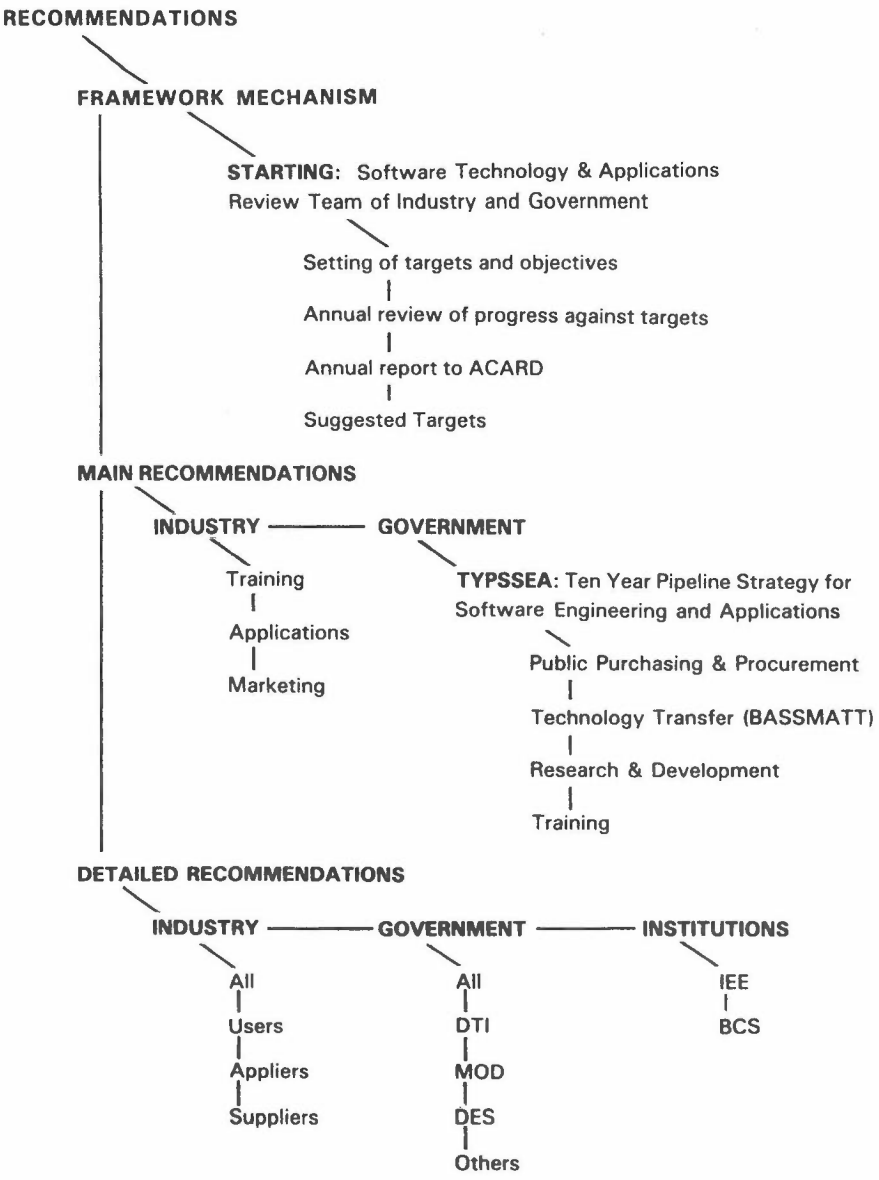


Figure 1. Structure of the recommendations

2. Introduction to IT, software and software engineering

2.1 Information Technology (IT) today affects almost every household and occupation in the United Kingdom. IT systems are at the heart of the financial services, the design of automobiles, the control of weapons systems, communications and many consumer goods.

2.2 IT is a combination of microelectronics, digital computers, software, computer programs and telecommunications. The components of an IT system have to be brought together by systems engineering, and computer programs must be developed to enable the components to work, both individually and together, as a whole system.

2.3 Computer programs (software) are the sets of instructions, stored inside a computer's memory, which when fed with the appropriate data, control the computer (hardware) in a way which causes the computer to perform some desirable function such as predicting tomorrow's weather or computing the magnitude of an overdraft.

2.4 'Software is the entire set of programs, precedures and related documentation associated with a . . . computer system.' (1) Software engineering is the collection of theories, methods, skills and tools used to produce software.

2.5 Computer programmers express their programs in a form which is readable by humans as well as computers. This is called 'source code' and is the generic name for programs expressed in such programming 'languages' as COBOL, FORTRAN, BASIC, PASCAL and ADA. Computer software in source code form (readable by humans) is analogous to an engineering blueprint. The actual value of a blueprint in terms of its physical components (paper and ink) is negligible. Blueprints, and software, are valuable because they capture the expertise of the designer who produced them; this expertise enables the end product to be manufactured.

2.6 The engineering blueprint needs a manufacturing process to realise the end product it represents. The 'end product' for software is called the 'binary form'

because it consists of the '0's and '1's which are stored in the memory of a computer. It is this pattern of '0's and '1's which actually controls the hardware. Such a pattern is 'soft'; it cannot be seen or touched. It is this ethereal nature of software which makes it difficult to grasp exactly what software is, what it does and why it is so important.

2.7 In the same way that a blueprint can be readily reproduced by copying so can computer software. The key difference, however, is that turning a blueprint into an end product and mass producing that product is usually an expensive and time consuming process. With software the transformation from design (source) to product (binary) is done cheaply and quickly by the computer itself! It also explains why it is so easy to make changes to software – changing the design is just text processing, and so a revised product is simple and cheap to produce.

2.8 Mass production of software is as easy and cheap as reproducing blueprints. Thus 'piracy' of computer software, the illegal copying and distribution of someone else's software, is very easy to achieve and is widespread in certain areas of the world today. This explains why vendors are often reluctant to distribute the source code of their products.

2.9 In industrial terms, the development of software is a new activity. Today it is regarded as a 'craft activity' because it is labour intensive with programmers using craft level methods. Most programmers have not been professionally trained. They do not usually have a deep mathematical training but have developed skills 'by experience'.

2.10 Software engineering is the application of science and mathematics, blended with experience, to the development of software. The theoretical basis of software engineering, while still in its infancy compared with some other engineering disciplines, has made steady progress. The pace of theoretical development is accelerating as the required mathematical principles are discovered and refined. The mathematics-led transition from a craft based industry to a mathematics based technology is gathering pace rapidly.

2.11 'Software can be divided into two generic types: systems and applications. Figure 2 shows examples of these types of software. Systems software helps the programmer by automatically controlling or managing the resources of a computer system. Applications software assists the user directly by performing specific tasks such as accounting or computer-aided design. Applications programs can be further divided into industry-specific and cross-industry.

2.12 'Another perspective is the mode of delivery of software to users: packaged, bespoke (custom) or integrated systems. Packaged software generally refers to a

standard program that can be used by a wide range of users with little or no modification. By contrast, custom software or bespoke programming is tailored to meet a particular user's need. Development of this software may take place on the user's premises or at the supplier's site on a contract basis. Integrated systems software is sold by systems integration firms which buy hardware from outside suppliers, add their own software and sell a complete system to users.' (2)

2.13 Participants in the IT business may be divided into three types: the user, the applier and the supplier:

- ⋆ *The user* is someone who employs an IT product to help him perform a task, but does not participate in the product's construction. For example, an accountant handling figures via a spreadsheet program is a user.
- ⋆ *The applier* does not consider the construction of an IT product as a primary task. Appliers will typically tailor some general purpose product, such as a spreadsheet package or a differential equation solver, into a more specifically useful tool by exploiting or 'applying' their knowledge of, say, the firm's accountancy procedures or the dynamics of an aeroplane. Appliers will have skills ranging from a simple understanding of the general tools (an experienced user) through to comprehensive programming skills (an 'internal' supplier).
- ⋆ *The supplier* is primarily interested in the construction and development of software for users, appliers or suppliers themselves. The supplier should have a high level of technical skill.

2.14 Users and appliers are typically found in the manufacturing and service industries. Large companies and organisations will often have their own specialist computing or data processing departments acting as an 'internal' supplier. The external suppliers are formed from the independent software houses and the computer hardware vendors.

2.15 Users, appliers and suppliers form an international 'vertical' market segment. The nature of their interaction and interdependence from the UK viewpoint is now considered, and some of the problems which we have identified are described.

Generic Type	Specific Type	Examples
SYSTEMS	Operating Systems	MVS/XA (IBM); CP/M (Digital Research); MS/DOS (Microsoft); UNIX (AT&T)
	Database Management Systems	ADABAS (Software AG) IDMS (Cullinet) IMS (IBM) RAMIS II (Mathematica Products Group)
	Telecommunications Monitors	ENVIRON / 1 (Cincom Systems) Softerm (Datasoft) VisiTerm (VisiCorp) CICS, TSO (IBM)
	Translators	Job Control Language Translator (DASD) Fortran Cross Assembler (National Semiconductor)
	Utilities	Supersort (MicroPro International) Program Editor (Radio Shack)
APPLICATIONS	Cross - Industry	Lotus 1, 2, 3, (Lotus Development) Peachpak 4 Accounting Series (Peachtree Software) VisiCalc (VisiCorp) Word Star (MicroPro International)
	Industry - Specific	Life - Comm (Informatics General) BANKSERV 10000 (Anacomp), MUMPS - 1; (Digital Equipment Corp) Milliken Math Sequences (Milliken Publishing)

Figure 2. Types of software (2)

3 The problem

A challenge to the UK's competitive position

3.1 The UK manufacturing and service industries (as well as defence, utilities and public administration) are now dependent on Information Technology (IT).

3.2 Almost every aspect of business in the UK is open to international competition.

3.3 In the economic struggle between America, Japan, the Far East and Europe, the application of IT is a major weapon.

3.4 IT is critically dependent on software.

3.5 The UK software market is dominated by foreign suppliers. This has three consequences:

a. The UK can no longer be sure of obtaining the software its manufacturing and service industries need to sustain their international and domestic competitiveness.

b. The increasing demand for imported software will bring about a major new source of balance of payments deficit.

c. The dominance of the UK and world markets by overseas suppliers is weakening the UK software industry. The UK industry is not effectively fighting back.

3.6 Should the UK be unable to afford, or not be allowed, to import foreign software needed to remain competitive, then the UK software industry will not be able to deliver competitive alternatives and so all manufacturing and service industries will be weakened.

3.7 While much ground has already been lost, prompt action can recover and improve the situation.

Software: a crucial component in the competitiveness of the manufacturing and service industries

3.8 A basic question which we have considered is whether the manufacturing and service industries can achieve and maintain their competitive edge if our indigenous software industry declines.

3.9 The dependence of the UK's service industries, manufacturing industries, defence and public services on IT has already been stated. The UK's immediate and future trading performance depends crucially on the combination of the country's ability to compete in world markets for products and services based on IT and on the rapid, widespread and effective application of such products and services by industry and commerce.

3.10 The types of IT on which manufacturing industry is necessarily becoming ever more reliant include computer aided design and manufacture (CAD/CAM), robotics and office automation. Increasingly the knowledge, experience and skills which make up a company's competitive advantage are being expressed as, and encapsulated into, software. Companies which do not lead this trend will suffer because such software will not be available for purchase. Instead it will be a strategic asset of their more successful competitors, for whom it will represent a considerable investment which will enhance future dividends.

3.11 Service industries are even more dependent on deploying advanced IT products to stay competitive; financial services now are almost pure IT activities. Japan and the USA have ambitions to rival the UK in the financial sectors and they are the major developers of key IT products. They will use this lever to try to offer better services than London's financial centres. Market forces, possibly aided by restrictions on the availability to the UK of key IT products, could lead to the loss of some important service industries.

Foreign control of software used in the UK

3.12 Information Technology is central not to just UK industry but to UK defence, public administration and utilities. The world and UK markets for IT hardware are dominated by the USA and Japan, with the USA alone holding 70% of the world software market in 1983 and a predicted 75% in 1987(2). The domination of the world IT market by one American company, IBM, is a special factor. IBM's dominance means that its products set de facto world standards. Most appliers and suppliers must comply with these standards and most users must operate within the limits defined by these standards. Therefore overseas governments and companies

have the potential to influence major strategic aspects of the UK economy, defence and administration by manipulating the price or the availability of software.

3.13 Vulnerability to price increase is a consequence of lack of competition from UK software producers. A large part of the UK software market consists of the operating systems, compilers, database systems and communications packages supplied by the foreign hardware vendors who are mostly American owned. With very few exceptions, hardware vendors deliberately construct this software to be specialised to their own products. Customers build on top of this basic software and thereby become dependent on one vendor. The cost of this software, together with the cost of applications-orientated software built for use with it, and the costs of training staff to use it, ensure a complete dependency on the hardware vendor. The vendor may legitimately exploit this market dominance for commercial gain. Due to the scale of the market dominance, such software dependencies do give the foreign vendors a potential for strategic manipulation, beyond legitimate profit making, which is not controlled by normal market mechanisms.

3.14 The availability of software may be restricted in at least three ways to influence the UK strategically: temporally, totally or partially:

a. Temporal restrictions refer to the potential for foreign suppliers to only sell 'last year's model' in the UK. This could ensure that the UK had a reduced competitive ability in important fields and markets.

b. Total restrictions refer to deliberate decisions to prevent the UK purchasing essential software. For example, the USA's recent technology embargo has demonstrated that this tactic could be used to gain commercial advantage.

c. Partial availability, such as supplying programs in binary but not source forms, is a common commercial practice. Strategically it means that, should a foreign supplier withdraw support and service from the UK, then the UK would not be able to maintain or develop the software on its own.

3.15 The UK, like many other countries, is thus dependent on American owned, developed and controlled software for the operation of most of the UK's computing power.

The IT balance of payments deficit

3.16 The UK balance of payments deficit for IT (including software) was £928 million in 1983 and is predicted to worsen to £9 billion by the early 1990s, of which software will contribute a £2 billion deficit (3). Figure 3 shows UK IT compared to other major UK loss makers.

3.17 Almost every aspect of business in the UK is subject to international competition. IT has intensified this competition. There is no escape from this competitive pressure. UK industry must enthusiastically and skilfully exploit the opportunities that IT offers or the UK economy will suffer.

3.18 The UK economy is dependent on IT and IT is dependent on software. Therefore software is critical to the economy.

3.19 The importance of software is reflected by the world market which is estimated at $40 billion for 1985, with an annual growth rate of 30%. The UK represents 5% of the world market. UK industry has only 2-3% of the world market; this share consists of about half of the UK domestic market plus only 0.5% of the rest of the world market.

3.20 UK software companies are growing at around 20% pa which is below the world rate of 30-40%. Thus the balance of payments deficit is going to increase if UK industry generally takes up IT at the rate required to remain competitive in world markets. This will drive the balance of payments deficit for software (all IT) from £200 million (£928 million) in 1983 to £2 billion (£9 billion in all IT) in the early 1990s. Two important questions are:

a. Can the UK afford such a deficit?
b. Can the UK afford to miss the opportunity which this fast-growing world market represents?

The pace and cost of software engineering

3.21 Now that the software industry is beginning to mature into a capital intensive, international industry, we are concerned that the cost and pace of technological development will be too expensive and too fast for UK companies to remain competitive.

3.22 The world market for computer systems continues to grow rapidly. Success in microelectronics technology means that software development now is the most costly and difficult component to develop of most IT applications, often amounting to 75% of the overall life cycle cost of a large bespoke application.

3.23 World demand for software continues to soar.

3.24 Software development is labour intensive because it is essentially all design effort and no production effort. Therefore, there exists massive competitive pressure to find new ways to improve the productivity and quality of the software

	£millions in 1983
Road vehicles	3669.6
Fruit and vegetables	1555.7
Paper	1363.2
Textiles	1035.2
IT (inc software)	928
Software	**200**
	Projected deficit for 1990
IT (inc software)	9000
Software	**2000**

Figure 3. Comparative trade deficits

development process; to find ways to move from a labour intensive to a capital intensive process. All the major nations and companies have large R&D programmes seeking these goals. R&D itself, the pace of technical change and the consequential costs of capital equipment and staff retraining, together with the even greater costs of marketing and supporting new products, are forcing up the price of remaining a world class competitor in the software industry.

3.25 This big R&D push towards more capital intensive methods of software development is likely to weaken the UK software industry. The UK is likely to under-invest in R&D, under-invest in the capital equipment, under-invest in staff retraining, and under-invest in development and marketing. There are too many small companies which cannot afford this investment and too many large companies who will not make the investment unless forced to by public purchasing pressure or government subsidy. Too many companies are reliant on government bespoke programming contracts (for example MOD work) which is somewhat sheltered from competition. Many such 'body shop' companies do not have revenue streams from software products to generate investment capital.

The future of the UK software industry in the 1990s

3.26 The indications are already present that the UK software industry is set to follow the pattern of the UK computer hardware industry (and others). Pioneering innovation created a new UK market for computer hardware which grew rapidly to support a number of small companies. These failed to match the market penetration, and later the technical strength, of their overseas rivals. The UK companies therefore grew more slowly than their overseas rivals. When the UK market became just a part of a vast international market, the UK companies did not hold viable shares of this larger market. After government-supported restructuring, through some unhappy mergers and takeovers, only one major UK company survives. The UK computer hardware market is now dominated by foreign suppliers.

3.27 Between now and the mid-1990s the UK software industry could evolve into one of three forms:

a. A 'net exporting' software industry contributing genuine wealth to the UK balance of payments and the economy generally. Other manufacturing and service industries benefit by exploiting the innovations of a dynamic, 'enabling technology' industry of world class.

b. A 'just enough' software industry servicing the needs of the UK manufacturing and service industries. This implies a major 'net importing'

software industry delivering foreign products to UK customers. A major balance of payments deficit on software products results.

c. A 'defence only' software industry, approximately keeping up with world developments, purely for strategic defence reasons. This implies a completely 'net importing' industry with little or no innovation by UK firms who are totally dependent on MOD business. It also implies that as the overall UK software industry declines rapidly it will lose the non-MOD business to foreign competitors operating in the UK market.

3.28 The 'man in the street' currently sees the UK software industry now and in the future as a high growth, export revenue-generating industry contributing significantly to UK employment and the balance of payments. This view is wrong. The recent USA study (2) confirms ACARD's view that the current trend is for the UK rate of growth to lag behind the world leaders, that the UK's world market share will decline, and that foreign companies and multi-nationals will dominate the world and UK markets in the 1990s.

3.29 The above decline will result from the action of current, international market forces. These include not only the normal commercial competition, but also political forces such as technology embargoes, import/export restrictions, tariff barriers, state subsidised development programmes and nationalistic procurement policies. The massive USA strengths of technical leadership, domestic market size, commercial vigour and investment, and government support will prevail in the world market. The remaining places in the world league will be taken by other countries such as Japan and France, whose commercial vigour and government commitment exceed that of the UK.

3.30 The UK domestic market alone is too small to sustain a world class software industry. If the UK industry does not more vigorously compete in the world market, encouraged by government action, then within ten years only MOD-supported firms will remain.

3.31 UK industry and government must together choose one of the options from para 3.27 and strive, jointly, to realise it. A 'window of opportunity' is still open but it is fast closing. The UK has the capability to build a long-term success in the software business. We ask whether there is the will and the energy in the UK to succeed.

4 Analysis: I. Software users, appliers and suppliers

The role of software

4.1 Since ACARD published its report on Information Technology in September 1980, the crucial role of software in IT has become even more apparent. It is now clear that software is vital to all sectors of the economy. The importance of software has been underlined, for example, by the difficulties encountered during the development of System X and the surveillance systems for Nimrod.

4.2 It must be emphasised that the service and manufacturing sectors are now highly interdependent. One aspect of the IT revolution has been to increase the common ground between various economic sectors. Some companies have seen this as presenting an opportunity to diversify their activities into new major areas of business.

The service industry

4.3 The service industry encompasses a spectrum of activities ranging from finance, banking and insurance; the retail trade; communications; entertainment; tourism and leisure; education; public administration through to the distribution and transport sectors. The industry is a major contributor to the country's wealth and now underpins practically all other sectors. Service industry exports, including invisibles, amounted to £19 billion in 1983. The service industry employs about 50% of the working population of the UK.

4.4 Part of the service industry has been involved heavily in IT for the past 20 years; for example, banking has 15-20,000 man years invested in software. Many service companies are totally dependent on IT. The service industries generate, by themselves, much of the applications software which controls their competitive edge and profitability. These applications, however, are usually based on hardware and systems software from the USA.

4.5 The consequential costs of software errors can be exceedingly high in sectors of

the service industry. These sectors require software which is of comparable quality to safety-critical applications (see Appendix B). They need the best software engineering techniques to achieve this quality level at a reasonable cost. Not all companies in these sectors are using widely the most modern techniques.

4.6 The productivity of the software development process is important to the service industries. Much of the software in use is a problem to maintain. Emerging business opportunities and awareness of IT benefits have created an accelerated demand for new and enhanced IT-based services. The maintenance problem and demand for new developments is placing an impossible burden on development resources and is causing a disappointing queue of unimplemented enhancements.

4.7 The ability to extend IT applications, without putting existing operations in jeopardy, is vital. This points to the need for an engineering approach to software production. Given the critical manpower shortage in IT, improved productivity of software development, by means of software engineering, is crucial to the continued good performance of the service industry.

The manufacturing industry

4.8 The current trend of UK market share for the industry is alarming; manufacturing recently has become a deficit for the first time. Reversal of this trend is dependent, however, upon an increase of productivity and marketing effectiveness. The widespread application of software to the manufacturing industry's operations, especially those of design, manufacturing and management, will play a key role.

4.9 To achieve the necessary gains in productivity, new manufacturing processes, advanced technology and traditional experience must be blended and encapsulated in reliable software.

4.10 Evidence shows that many overseas nations and companies have set productivity goals as high as the best UK manufacturing companies. There are strong indications that many overseas companies are well positioned to achieve their goals. The objectives of each industry and each company vary in detail, but their targets include:

* 50% reduction in timescale from design start to market entry for new products;
* 50% reduction in the cost to launch new products;
* major improvement in initial reliability of new products;
* increase in model diversity so as to improve customer choice and satisfaction, without incurring unacceptable costs;
* achievement of these objectives within a decade.

4.11 The central importance of design in manufacture is receiving belated recognition. Prediction of both unit cost and the cost to prepare a new product now can be settled at the time the design is released for manufacture. Soundly based software, capable of aiding design analysis and optimisation, is a key to attaining timely design objectives.

4.12 Improved modelling of manufacturing processes and work flow is vital to achieving greater production efficiency. Better methods of design analysis are vital to match customer needs, designs and process capability. A good match is a major factor in achieving high productivity and control of costs.

4.13 A major development in recent years has been that major companies, for example Boeing and General Motors, are willing to do business only with those companies, large and small, who are able to adhere to their IT standards (eg Manufacturing Automation Protocol (MAP) and electronic data interchange). Effective software capability is the key to being able to meet these requirements. Companies not keeping pace with the technological standards of major companies may find themselves unable to trade in these sectors.

World market for software

4.14 Software has become an international commodity. There are large, world-wide business, domestic and industrial markets for general purpose software. World markets also are open for specialised software. In this section, we consider the world market for software in relation to the UK software industry and the software industries of other countries.

4.15 An estimate by the USA Department of Commerce (2) put the world market for software in 1985 at $40 billion. Growth was estimated to be about 40% compounded annually (Figure 4).

4.16 The growth of the UK domestic software market is slow when compared to world market growth. The USA has 75% of the market, while the UK has only 3%. France, Germany and Japan each have about twice the share of the UK (Figure 5). Only about 7% of the UK-produced software is exported.

4.17 The world market currently is dominated by packaged software. The USA is dominant in such software, with packages contributing almost 60% of total USA software revenue, while the UK mainly produces bespoke, integrated systems. Computer users are buying more packaged systems to conserve their programming resources. The UK software industry would appear to be out of step with the trend of the world market.

4.18 Much of the market for software is 'invisible' because software often is embedded into products; companies develop applications software in the course of their business; and large numbers of people are engaged in preparing software for companies outside the IT industry. It is reasonable to assume that some of this effort is applicable to wider markets; it is important that the UK capitalises on this 'hidden' investment.

The UK software industry

4.19 The software business in the UK consists of suppliers, appliers and users. Many of the user companies are themselves important developers of software. A wide variety of companies are involved in the software business from very large multi-nationals to very small local firms. These companies encompass a myriad of approaches to producing software.

Suppliers

a. *Overseas computer companies*

4.20 At the top of the scale is IBM(UK) with revenue of £2.5 billion in 1984 (approximately 20% of that revenue was from software). Some 50% of total IBM(UK) revenues, including an estimated £200 million of software, is derived from exports. The size of this one company and the influence which it has on all software activities cannot be over-emphasised. IBM(UK)'s turnover matches that of the rest of the software industry. Consequently, consideration of the health of the UK software industry must include IBM.

4.21 Also of great significance in the UK are other very large, American-owned, multi-national companies such as DEC, Honeywell, Hewlett-Packard, and Wang. DEC and Hewlett-Packard have set up major software development centres in the UK. Revenue figures are not available as a whole, but DEC, for example, had a 1985 turnover of £430 million in the UK.

b. *USA software companies*

4.22 There are many software supply companies in the USA. The major product of these companies is packaged applications software, in contrast to software revenues of hardware suppliers which tend to be from packaged systems software. Many of the products sold by software supply companies in the UK are imported from such USA companies. These USA companies are increasingly operating directly in the UK.

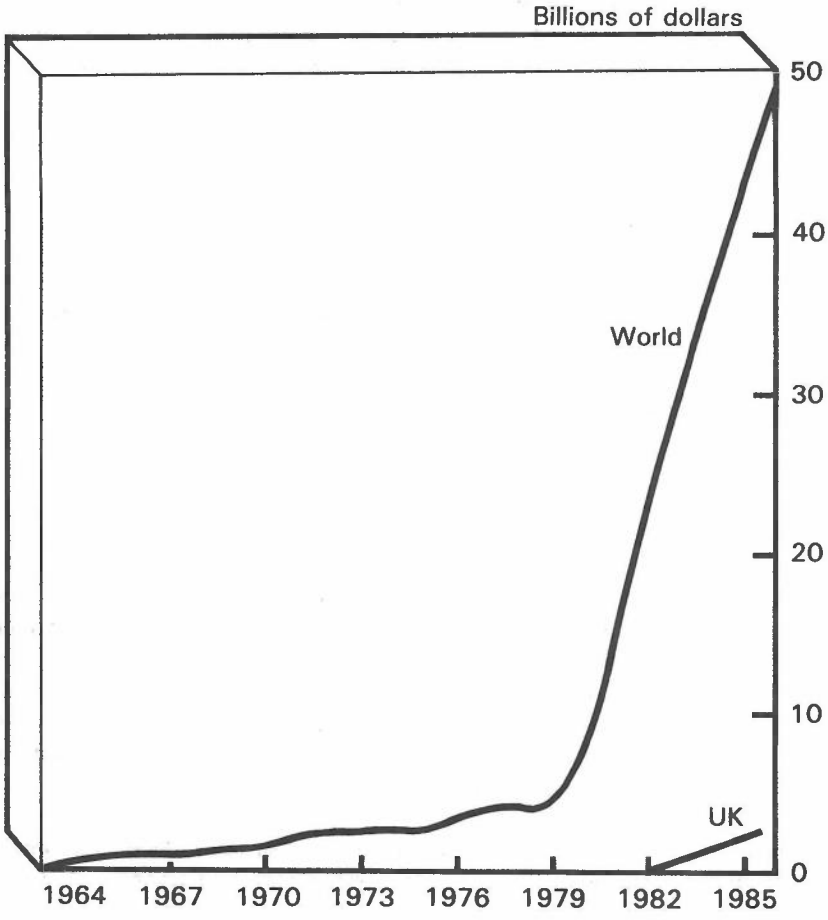


Figure 4. Estimated world market for software

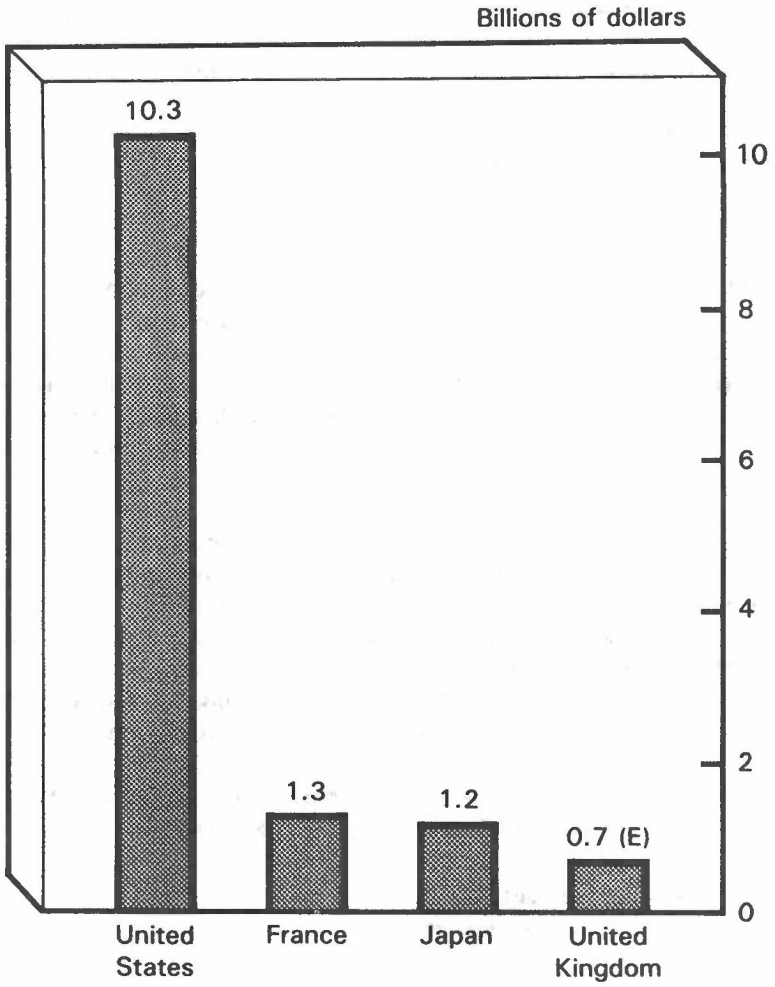


Figure 5. 1982 revenues of software industry by supplier nation

c. *UK computer hardware companies*

4.23 The UK has one major computer hardware company, ICL, which produces software for its range of hardware products. Recent years have also witnessed rapid growth of a number of other small hardware companies, based mainly on personal and home computers, for which a network of software supply firms has developed. A major consequence of the smallness of the UK hardware producer base is not only the loss of revenue on hardware sales but also on the systems software required to operate computer hardware.

d. *UK software houses*

4.24 The leading UK computer software service companies are CAP, Centre File, Compower, Hoskyns, ISTEL, Logica, Scicon, Software Sciences, Systems Designers and Thorn-EMI. In 1984, these companies each had revenues in the region of £20-60 million. Collectively, these companies often are identified as the UK software industry, although, as indicated earlier, much software is produced by companies not identified as part of the UK software industry or is imported from the USA.

4.25 A major structural problem of the UK software industry is its fragmentation into many companies unable to generate sufficient revenues to undertake the development of marketing of new software products. Because of their small size, UK software houses are vulnerable to acquisition by major multinationals; two major UK software companies are now USA owned and more are likely to be acquired. We suggest that the sector is ripe for rationalisation into larger entities which are able to compete better with multinational companies.

4.26 The Computing Services Association, which includes most of the UK software houses, reported that in 1983 the computing services industry derived its revenues from:

28% processing (bureau)
18% bespoke (custom) software
18% total systems
12% consultancy
11% software products
13% other software

About 14% of the revenues of the software houses are derived from overseas sales.

4.27 The UK software industry has much to be proud of. It is growing faster and more profitably than almost any other sector in the UK economy (Figure 6). Growth still is too low, however, if it is compared to the performance of the

world software industry and the potential of the market to sustain higher growth. A combination of dynamism and protectionism by the USA, Japan and other nations poses a genuine threat to improvement of the UK's share of the world market. All of this emphasises the need for the UK software industry to rise to the challenge.

4.28 For the UK software industry to be profitable in the face of international competition it must:

- ⋆ offer competitive prices, estimate development costs accurately, and deliver software on time and to budget – otherwise it will fail to win orders for bespoke developments or lose money on those orders it does win;
- ⋆ deliver software of high quality – otherwise it will fail to sell its products and fail to win orders for bespoke developments;
- ⋆ respond quickly to changes in market demand for products.

4.29 Another problem of the UK software industry is that it does not collaborate sufficiently with hardware suppliers. In the USA, software companies keep close contact with computer hardware producers and, therefore, are in a position to offer ranges of products for new hardware in a timely fashion. UK software companies do not have this geographical advantage and one result is the heavy importation of USA software.

Appliers

4.30 The UK public and private sectors are customers for the software houses but are also major developers of applications software. Large electronics companies, such as GEC, Plessey, British Telecom, are a significant national force in the development of software. Other companies, for example Austin-Rover and BP, are significant spenders on software and have been instrumental in spinning off major software companies. Many UK companies appreciate the wider significance of their applications software but often have problems marketing it outside their normal operations.

4.31 Increasingly, companies with the capability to produce applications software in-house distinguish between strategic software which is necessary for their competitive edge and software of general applicability. General purpose software often can be developed more cost effectively from pre-written modules and packages than from 'scratch'. Opportunities exist for the software industry to produce these modules and the vital interface software.

Users

4.32 Practically all companies and organisations are software users. In the USA,

major hardware and software companies show keen awareness of the needs of the user community in developing new products; for example, user groups and conferences are an important feature of the USA industry. We consider that the relationship between user and supplier in the UK is not sufficiently close and recommend measures to rectify this problem.

4.33 In the case of small users of software, ignorance of methods for specifying, implementing, operating and maintaining systems is contributing to failure to benefit from small systems; this could become worse as users are driven to using more complex systems. The issues relating to the small producer and small user are, in some respects, a microcosm of the problem facing the whole industry. We commissioned a study by the National Computing Centre to focus on the issues relating to small users and suppliers of software. The NCC made a number of useful suggestions and proposals which we now urge them to follow up.

Customer factors limiting innovation

4.34 The recent PREST report (**4**) about acceptance of new technology argues that 'the structural and industrial implications of new information technology will be dependent on the willingness of consumers to accept the services they provide. It is apparent that acceptance is a gradual process in which consumers evaluate the performance and economic characteristics of IT.

4.35 'It is one matter for consumers to accept new IT products and services. It is quite another matter for these to be produced profitably by UK entrepreneurs. Acceptance is not a matter of consumer behaviour alone, but also a question of a balance between demand and supply. Effective supply depends upon a complex relationship between infrastructure, equipment and information services. The cluster of technologies and services which constitutes new technology contains within it several alternatives for satisfying latent needs for entertainment and information services.

4.36 'A number of policy perspectives may be deduced:

- ⋆ *There is a need to stimulate a high rate of experimentation in the IT field* In this context, the market mechanism is not only a means of allocating resources but also a framework for encouraging entrepreneurship by both consumers and producers. The high risks faced by initial entrepreneurs need explicit recognition as do the barriers to exploitation created by high and indivisible infrastructural costs. A policy of subsidy or support for experimental pioneers is necessary.

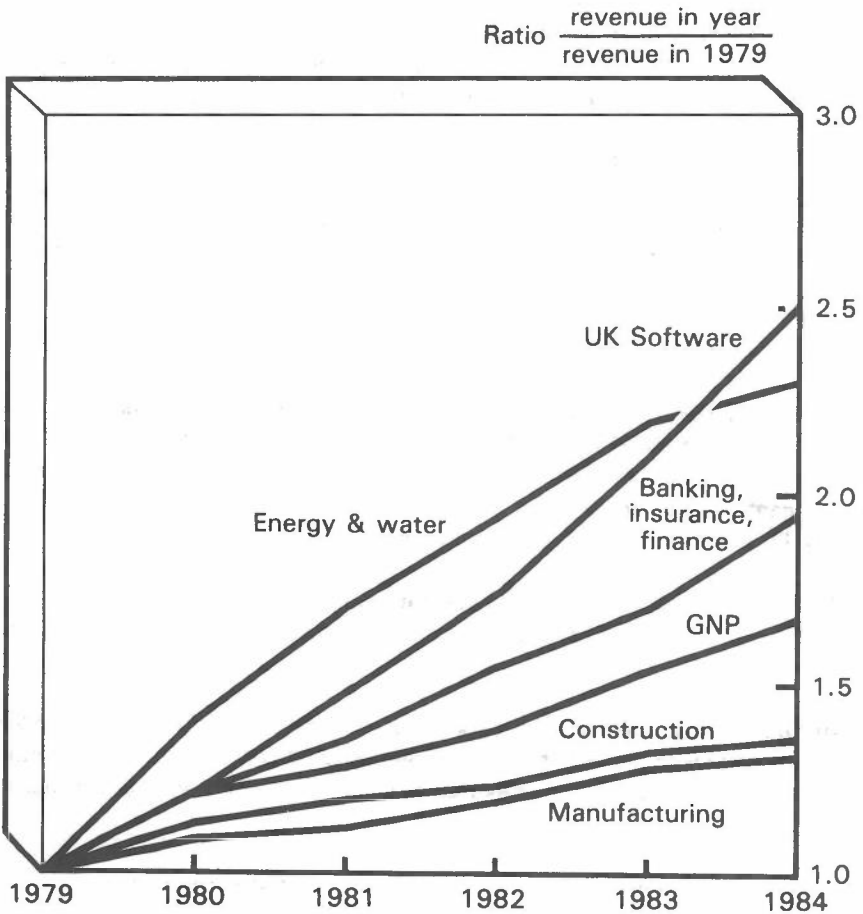


Figure 6. Growth of UK software industry compared with other sectors

* *Knowledge acquired by the consumer is the key to rapid acceptance*
  Much can be done by Government to create awareness of IT services and technological opportunities. The analysis contained in this report suggests the importance of identifying the needs of key innovatory groups and targeting campaigns specifically at them. A principal determinant of the relevant adopting population is the match between the performance requirements they impose upon a new technology and its level of technical performance.

4.37 'Flexibility of policy should be the overriding concern if a framework for market experiments and consumer acceptance is to be encouraged. Policy makers must recognise the uncertainties inherent in technological competition to no lesser extent than the entrepreneurs on whose initiative the development of functional areas depends.' **(4)**

User-applier-supplier symbiosis

4.38 If a market is driven by technology push (supply side leadership) and customer pull (demand-side leadership), then the rate of growth innovation and application of innovation is driven by the technical expertise and vision of all three participants – users, appliers and suppliers. The UK has been slow to take up and apply IT innovations, even those which have been invented in the UK.

4.39 There is a need for improved communication and feedback between users, appliers and suppliers. The rate of circulation and spread of new ideas must be increased.

4.40 Only a technically aware market can demand innovation to set a pace commensurate with the rest of the world. Successful IT companies in the UK must be technically innovative, and supported by equally innovative, market demand at home.

4.41 Software companies need to become more product orientated. Market targeting requires symbiosis between users, appliers and suppliers.

The United States of America

4.42 The USA software industry dominates the world; in 1983, it held 70% of the world market. Its leadership appears to be due to the predominance of USA hardware manufacturers, the entrepreneurial culture of the nation and the presence of a large internal market. USA hardware and software industries have restructured into large companies although there remain a significant number of small, independent companies. The small companies are constantly in flux; this

may help maintain a dynamic industry which is responsive to change and willing to exploit innovations.

4.43 The USA software industry sees major opportunities for expansion. In spite of their dominance, they are making massive software investments in response to perceived threats from abroad, mainly from Japan.

4.44 Major USA software research and development initiatives have been established with public and private funding. Some are:

* The Microelectronics Computing Corporation (MCC), formed by twelve companies who have provided $120 million over five years, for a 15-year advanced software engineering programme;
* The Software Productivity Consortium (SPC), with about the same budget as MCC, formed by twelve aerospace and defence companies to achieve technology transfer;
* The DOD Software Engineering Institute (SEI) based at Carnegie Mellon University and due to last at least 15 years with an initial $100 million over five years. SEI is 100% supported by the Department of Defense (DOD) and the major objective is technology transfer to DOD's suppliers; spin off into the civil sector is expected;
* The Strategic Computing Initiative which funds long-term research;
* The ADA Joint Program Office which is concerned with promoting and controlling the ADA language;
* STARS activities which relate to software engineering technology for producing real time systems;
* The Strategic Defence Initiative (SDI) which will inject massive funding into computing and software.

Japan

4.45 Japan is following a strategy in software broadly similar to that used to achieve success in other markets, such as consumer electronics. Emphasis is placed on product enhancement in well-defined sectors; quality, image and competitive pricing are essential components to their products. Although the Japanese software industry started late, it is now the third leading supplier of software in the world.

4.46 A characteristic of Japanese software research and development is partnership between government and industry. Duration of programmes is sufficiently long (10-15 years) to allow transfer of the results to industry; much of the research is carried out by subcontracting to industry. Japan has injected considerable public funds into software research and development, for example:

* The fifth generation computing project has a budget of $450 million over ten years;
* The very high speed scientific computing systems project started in 1981 and will receive $200 million from MITI over five years;
* A major $100 million technology transfer programme, called SIGMA, to provide a national integrated project support environment for software producers.

France

4.47 The French software industry has become the leading software supplier in Europe and second in terms of world market share. Efforts have been made to ensure that the industry is export orientated and a quarter of total revenue is from export; the USA is their major customer.

4.48 An incentive to the software industry in France has been proposed by Government. This will allow increased depreciation of hardware used in the development of software and the accelerated write off of R&D expenditure.

4.49 Under the French five-year Electronics Plan/Software Program, which started in 1983, $685 million will be provided by the French Government for research and development. The three year, $15 million, Mantis project began in 1983.

West Germany

4.50 The software and services market in West Germany was the second largest in Europe in 1984 at $2.4 billion. There has been a shift from bureaux services to software and systems; a number of new software houses have been established in this dynamic sector. Packaged software was the fastest growing sector at 34% accumulated growth in 1984 and accounted for 40% of the total software market.

Sweden

4.51 The Swedish IT industry was rationalised by the Government in the 1970s with subsidies of 450 million Swedish kroner. Reorganisation of the industry, identification of niches for exploitation and co-operation between government and industry have allowed effective competition with foreign competitors. For example, Sweden now is a world force in telecommunications.

4.52 In 1982 and 1984 the Swedish Government funded public computer awareness

and literacy campaigns. A scheme was announced recently to allow 10,000 people per year to receive basic education in IT.

EEC

4.53 The countries of Europe have discussed ways of increasing collaboration to overcome the problem of fragmentation of the European industry and market. This is embodied in the Strategic Programme for Research and Development in Information Technology (ESPRIT). ESPRIT began in 1984, will end in 1989 and has a total budget of $645 million; $100 million has been earmarked for software development.

4.54 A European computer research institute was established recently in Munich through the collaboration of companies in the United Kingdom, France and West Germany. Initial funding is $15 million. Work will focus on software and networking techniques for fifth generation computer systems and is intended to complement ESPRIT.

Other countries

4.55 It is known that other countries, for example, India, Korea, Taiwan and Singapore, are seeking to establish a presence in the world software market. A major advantage of such countries is lower labour costs in an industry which presently is labour intensive.

5. Analysis: II. Technological development

The engineering approach

5.1 Software engineering is the application of sound scientific, mathematical, management, and engineering principles to the production of correct programs, on schedule, within estimated cost and at a competitive level of performance and price. An engineering approach to software must address all functions which relate IT to market needs and product support. Dependency on software is increasing in all sectors of the UK. Consequently, sound engineering practice, at all points in the software life cycle, is vital.

5.2 Software production is a difficult process to manage. Acceptable metrics for the productivity of software generation or for the quality of software products have not been devised. The difficulty of management reinforces the need for a sound engineering approach to software.

Quality

5.3 Computers are machines with a theoretical capability of performing perfectly. Their poor performance, in terms of quality and reliability, stems from the software which controls their behaviour. Software quality demands:

* fitness for purpose
* reasonable cost
* reliability
* ease of use in relation to those people who use it
* desirable maintenance and upgrade characteristics
* compares well against rival products.

5.4 High quality software presently demands labour-intensive methods, such as formal review and exposure to the collective wisdom of experienced practitioners. Similar practices are essential in other engineering activities, and their systematic

application to software production would do much to improve software quality. Increased use of the best experience-based techniques is vital.

5.5 Increasing use will be made of software modules and packages which will be integrated into specialised systems. Quality at this level is essential to ensure the quality of systems based on these components.

5.6 Mathematics promises a more rigorous and cost effective means of producing high quality software. The amount of science and mathematics which can be used to supplement human experience and reduce dependency upon human judgement is increasing. Many of the precise mathematical concepts appropriate to computer programming have been identified to the point where some classes of programs can be generated with the rigour of pure mathematics. The use of mathematically based techniques is likely to increase significantly in the next ten years as accelerating research and development increase the rate of technology transfer. The take up of these techniques will be most profitably exploited by those companies who prepare their staff, in advance, with the appropriate mathematical education.

Productivity

5.7 In human terms, the investment in software is large; banks, for example, already have invested about 20,000 man-years of effort. With the existing burden of software maintenance and explosive growth in demand for new applications, massive improvements in productivity are required. The increasing cost of labour and shortage of skilled personnel suggests that requirements cannot be met by growth of staff.

5.8 Apparent increases in productivity are occurring already as users move to packaged systems, prototyping, 'fourth generation languages' and modular software. These approaches do not address productivity over the entire life cycle of software.

5.9 Significant and genuine increases in software productivity can be achieved only by an engineering approach.

5.10 Methods and tools have been created to assist software engineering; they are essential in large-scale software projects. Benefits of using methods and tools, in any size project, are evident when specifying, producing, verifying and modifying systems. Methods and tools incorporate notations for describing the system, guidelines for producing descriptions and rules for verifying consistency and accuracy.

5.11 Current industrial use of methods and tools is not adequate. Many producers and users would benefit greatly from the immediate introduction of existing methods and support tools.

5.12 Integrated project support environments (IPSEs) (**5**) are a practical embodiment of accepted software tools and methods. The development and use of IPSEs should be encouraged. The IPSE concept is neither widely understood nor applied in practice. We believe that in-service education and training will help to redress this deficiency and that IPSEs should form part of the UK's software research programmes.

5.13 Further automation of the development process is attractive as a valuable means of making better use of the available talent. This can be achieved by interactively checking the design details, and providing the designer with immediate analysis of the consequences of his design decisions. Important areas are mathematical techniques, measurement of software quality and estimation of software costs, timescales and performance. These should be addressed by the UK's software research and development programmes.

5.14 Software developments and products must be documented. Correct, formal and complete documentation does not always receive sufficient attention from software developers. We believe that in-service education and training will correct this problem, in part. Tools for documentation should form part of the UK's software research and development programmes.

5.15 The Software Tools for Application to Real Time Systems (STARTS) initiative is intended to improve the quality of complex real time systems and to improve productivity of their productions. We believe that STARTS does not include enough participants from industry and government.

5.16 We welcome the work of the STARTS tools evaluation exercise but believe that this activity needs expansion. More cost-benefit studies are needed of the use of advanced software engineering techniques.

Skills, education and mobility

5.17 Lack of understanding and awareness of software engineering principles in industry, at the highest levels, is a serious concern. 'Some 10-20% of business expenditure (7% of GNP) is now for computers and software . . . their optimum use is highly influential in the success or failure of an enterprise' (**6**).

5.18 We believe that widespread lack of competence in IT is a major cause of many

of the problems associated with software in the UK. All staff involved with IT need a coherent appreciation of the subject. Experience, while absolutely necessary, is not sufficient in a technical field such as IT; education in the theoretical basis of the subject is necessary.

5.19 The UK has a well-known shortage of IT skills at all levels (**6**). There is not time to wait for trained recruits to be produced by the education system. The only way to alleviate the shortage in a useful time frame (that is, 1990) is to educate and train members of the current IT population.

5.20 The UK tends to be poor at in-service training. Many firms are not prepared to spend any money on training; typical firms invest no more than one or two days per year per employee. In-service training tends to be only of immediate applicability.

5.21 Continuous and rapid technical change in IT means that those involved in the field require regular in-service education and training. Poor or obsolete technical skills and poor general IT awareness hampers not only the supply side; there is evidence to suggest that the demand side is constrained because users do not have the knowledge to exploit IT fully.

5.22 Employees need experience enhancement, as well as formal education and training. There is not enough interchange between academia, government service and industry. Mechanisms such as secondment and joint projects, used extensively in the Alvey Programme, are showing the benefits of this form of mobility.

5.23 Another skills problem in the UK is that of employee immobility. There is an underlying cultural reluctance to change jobs; this is damaging to a highly innovative industry. Immobility makes it difficult to obtain skills which may be available nationally, but not in certain areas, and makes it difficult for individuals to avoid technical insularity. Portable pensions, secondment and collaborative projects help reduce immobility.

5.24 It is essential that management understands the necessity of training and education in the IT field. Our message to management is:

- Development of software, in terms of education, requires significant capital investment.
- Computer professionals require frequent re-training because of the high rate of technical progress in software engineering.

5.25 The need for an engineering approach to software necessitates a fundamental change in approach which is not appreciated completely within the educational

system. All levels of the system should emphasise software engineering in computer education.

Standards

5.26 Regulations and standards may promote or inhibit growth of IT. Formulation of standards is a careful, detailed, tedious and time-consuming process; often it is perceived as having little commercial value. The importance of standards often is underrated and the most competent people are not always involved.

5.27 Standards are vital to the UK IT effort. Other countries appear more adept at exploiting standards than the UK. This is done by promulgating national standards as international ones or by impeding foreign technology with counter-standards.

5.28 Commercial implications of standards can be considerable. Suppliers and users gain competitive leverage by intelligent and timely adoption or imposition of standards. Education can play an important role in achieving this goal.

5.29 Standards are a major technology transfer mechanism and contain considerable research content. Not enough support is given by the research community to standards nor do standards receive sufficient emphasis in educational curricula.

5.30 Increasing use is likely to be made of reusable software components which will be integrated into systems. Standards will be needed to ensure compatibility between software components and acceptability in the marketplace.

5.31 Support from all concerned is needed to ensure that UK representation is technically and politically successful in the standards arena. Industry must be prepared to play a leading role in standards-making. Government must ensure UK strength in the international standards arena. Academics and researchers should be encouraged to participate in the development and teaching of standards.

5.32 Many IT standards have been based on a mixture of merely empirical knowledge and commercial self interest. Future standards should be based upon firmer scientific and mathematical bases, and with more consideration for the user of the standard and less for its supplier (see discussion in Appendix C).

6. Analysis: III. The role of government

The public pipeline

6.1 The UK government plays an influential role in all aspects of IT. Goverment influences:

- ⋆ national IT policy
- ⋆ national targets and expectations for IT developments
- ⋆ national awareness and attitudes
- ⋆ education and training
- ⋆ public research and development
- ⋆ public purchasing and procurement
- ⋆ sponsorship and support of industry
- ⋆ national and international regulations and standards
- ⋆ the legal framework for the applications of IT.

6.2 Responsibility for these roles is split between many government departments, including DTI, MOD, DES, DHSS, Home Office and CCTA. The IT roles played by government can be viewed as a pipeline which models the software industry (Figure 7).

6.3 Public funds finance almost all of the long-term research in software engineering and novel applications. Thus government, directly or indirectly, controls the feedstock of the IT industry's pipeline. Much mention is often made of the rapid pace of innovation in software and IT. Some products often only have a lifetime of two to three years before they are superseded by some new innovation. It should be realised, however, that it often still takes 10-15 years for a novel idea to move from original conception to the product stage. The rapid product evolution is fuelled by the international scale and pace of the continuous research and development pushing through into the product domain.

6.4 This pipeline and the skills and infrastructure it represents take many years to build up; they need constant maintenance. If the flow is broken, by 'stop/go' funding of research, for instance, then gaps in the product stream will occur. Such gaps cause firms to lose markets. These cannot be recovered because it is either too expensive or impossible to win back these markets.

6.5 The need for long-term continuity in research and development is recognised by America and Japan. In the UK there is a history of 'stop/go' funding and a fatal tendency to stop funding too soon after the initial stage of research. The UK tends not to push an idea far enough into development so that industry can have a suitable demonstration of the idea's worth, in order to justify further product development investment, or declare the idea demonstrably unsuitable for exploitation.

6.6 The Alvey Programme has been a major step forward in bridging the gap between academia and industry, between research and development. In 1983 Alvey was a 'go' force. Due to a lack of forward planning the momentum generated by Alvey is about to be lost as Alvey cannot fund new projects from 1986 onwards. Thus the feedstock of the pipeline is being 'stopped'. This is yet another example both of sporadic government involvement and of UK inability to push research into development. Such discontinuities are extremely harmful to UK research and industry generally.

The development-procurement gap

6.7 Government departments who sponsor research and development, such as DES and DTI, do not have major IT procurement requirements. Those departments who procure large quantities of IT, such as MOD and DHSS, do not sponsor much research which is not focused on specific departmental requirements. There is no continuity of departmental support or involvement through the 'research and development to customer' pipeline. This discontinuity means that no one person or agency can 'champion' an idea from its conception through to commercial demonstration or availability. The formal and informal barriers between departments constitute the well known research-development and demonstration gap; they also form a development-procurement gap (Figure 7). This renders impotent a major lever to help the IT industry; namely the imaginative use of public procurement to pull new ideas into the market-place.

6.8 The Government's lowest compliant bid philosophy works against suppliers risking innovation, because innovation requires the costs of investment in tools, methods and training. Thus, stagnation and risk aversion pervade those sectors of the software industry which are dominated by government procurement.

6.9 Lack of IT appreciation and skills is worse within government than it is in industry. This tends to make many government attempts to support industry ineffective or even counter-effective.

6.10 There is neither widespread perception of a coherent UK IT policy nor of national targets. Views differ from forecasts of unlimited growth to forecasts of the UK becoming a 'slave' of international IT suppliers. National awareness of IT appears to lag behind other nations, in spite of action taken by government departments in recent years.

6.11 The UK's competitors are raising the stakes in the competition for the world IT market. Massive foreign government research and development programmes are backed by sensible public purchasing policies, effective technology transfer programmes, publicly assisted marketing and better educated populations. In the face of this competition, the UK must respond decisively if it is to meet these challenges.

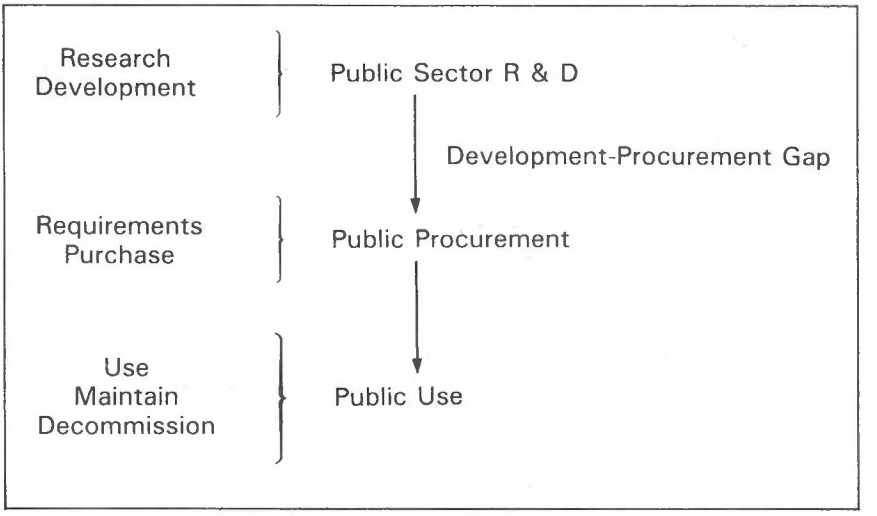


Figure 7. The public pipeline: development-procurement gap

7. Recommendations

STARTING: Software Technology and Applications Review Team of INdustry and Government

7.1 The widespread application of soundly engineered software has the potential to enhance the national competitiveness of the UK. In this chapter we make our main recommendations which we believe will maximise this potential. These main recommendations are divided between industry and government. In the three subsequent chapters we give more details about these main recommendations and make further recommendations. Chapter 8 contains the detailed recommendations for all industries and for the user, applier and supplier roles specifically. Chapter 9 contains the detailed recommendations for government; these are assigned to individual government departments. Chapter 10 contains detailed recommendations for the professional institutions.

7.2 Software is so important to the UK that the implementation and effectiveness of our recommendations must be regularly monitored against clear targets which we have provided. The organisations concerned must have a mechanism by which the recommendations and targets can be communicated. This mechanism must be responsible for performance evaluation on an annual basis. Feedback from the organisations must be part of the mechanism to allow adjustment of the recommendations and targets according to changing realities.

7.3 **We recommend** the formation of STARTING – the Software Technology and Applications Review Team of INdustry and Government. STARTING should:

- ⋆ be comprised of outstanding individuals drawn from industry and government and academia, representing users, appliers and suppliers.
- ⋆ submit to ACARD an annual report outlining progress towards the established targets and explaining any intentions to modify targets.
- ⋆ convene a well-publicised, plenary annual general meeting of two days duration. The objectives of this meeting should be:

- to provide an annual focus of publicity on the national importance of software;
- to report to the public on the past year's performance and overall progress towards the established targets;
- to communicate targets for the next year and the next ten years;
- to invite papers and demonstrations pertinent to the goal of improving the UK's competitiveness by the development and application of software and software engineering;
- to provide an opportunity for formal and informal contacts between those involved in all aspects of software with a view to improving the user-applier-supplier, vertical market symbiosis and increasing the efficiency of the research-to-procurement public pipeline.

7.4 **We recommend** that the initial targets set by STARTING be:

* that the UK should have a software balance of payments surplus by 1995;
* that UK-owned companies should, by 1995, supply:
  - 80% of bespoke/custom software produced in the UK
  - 50% of the new packages purchased in the UK
  - 30% of the systems software employed in the UK
  - 10% of the total world market for software;
* that companies operating in the UK should, by 1990, increase their formal, in-service, software training to:

Board and senior managers	1 week per year
Middle managers	2 weeks per year
Technical IT staff	4 weeks per year
Non-technical IT staff	2 weeks per year
Other professional staff	1 week per year

Main recommendations

7.5 Our main recommendations for industry are:

* in-service training initiative for all users, appliers and suppliers;
* increased application of software to increase the competitiveness of the manufacturing and service industries;
* marketing initiative for UK software products and services.

7.6 Our main recommendation for government is the formulation and implementation of a long-term plan (TYPSSEA) for inter-departmental co-operation on:

* public purchasing to exercise demand-side leadership;
* a new technology transfer initiative (BASSMATT);
* better R&D planning;
* in-service training initiative.

Main recommendations to industry

Training

7.7 The key to national success in both using and developing software is the effective education of those working with it. There is, at present, a low relevant educational attainment of most professionals involved in software. This problem has been recognised and measures have been taken to enable the UK educational establishment, in the long term, to meet the increasing needs of industry for staff trained in software technology and usage.

7.8 UK industry cannot afford to wait complacently for these recent changes in education to take effect. Industry must take responsibility for meeting its immediate needs. At present UK industry makes insufficient investment in training those staff involved with software. Lack of training investment is blamed upon rapid obsolescence of training, perception of low cost-effectiveness of training, high rates of staff wastage and those firms who 'feed off' the training investments of others. Without the increased competence, effectiveness and productivity which training can provide, UK industry will find itself increasingly unable to deliver the software which it needs to compete.

7.9 **We recommend** that all sectors of industry using software increase their provision for in-service training to meet the following annual targets:

* Directors, executives and senior managers receive one week of technical training.
* Managers receive two weeks of technical training.
* Technical IT staff receive three weeks of technical training; and one week of business and marketing training.

Further discussion of the technical, business and marketing training is to be found in Appendix A.

7.10 There are significant benefits to be gained from following this recommendation:

* Many managers do not appreciate the importance of software to their businesses

or understand the fundamentals of software; in-service training is a means of addressing this problem.

- ⋆ Many programmers do not have an appreciation of the fundamentals of business, marketing or the importance of software to industry; in-service training can correct this.
- ⋆ Many programmers and managers do not have an adequate mathematical and logical basis: unless this is remedied they will be unable to capitalise on the emerging software engineering techniques which are mathematically based. Such mathematical groundwork will be relevant for the rest of their careers and thus will bring a continuous long-term benefit.
- ⋆ The rapid obsolescence of technical and managerial staff, even those properly educated at one time, is unavoidable in a dynamic field such as software; in-service training is a means of addressing this problem.
- ⋆ The UK has a shortage of software skills at all levels; in-service training is a means for industry to 'reskill' existing employees to meet this need.
- ⋆ The scale of training requirements which the recommendations imply provide an opportunity to invest in producing cost-effective training packages and delivery mechanisms.

Applications

7.11 In contrast to the technically competent base of indigenous software suppliers, UK industry has been slow to apply software to its business activities. Consequently, industry has lost considerable opportunity to enhance its competitiveness. As long as UK software efforts are led by suppliers, industry cannot realise the full benefits of IT. Industry must take steps to ensure that it is able to take advantage of software technology by applying it appropriately.

7.12 There is a close dependency between suppliers and users of software. Without the timely availability of suitable and cost-effective applications software, many users will be unable to compete. Without user demand, software suppliers will not produce suitable products and, ultimately, will not prosper. It is unfortunate that UK software suppliers do not have effective collaboration with an indigenous, world ranked computer hardware industry. This means that they are not in the best position to offer new and timely software products based on new hardware. UK users are unable to capitalise on new hardware developments unless they turn to foreign software suppliers. Consequently, UK suppliers are in an inherently unfavourable competitive position. Because of this, application of software in specialist domains is particularly important to the health of UK software suppliers as well as users.

7.13 **We recommend** that the application of software by industry be accelerated. We believe that the following measures will be effective:

- ⋆ In-service training will greatly assist the circulation of ideas and help create a dynamic climate of software application; a major goal of the training programme should be a substantial increase in the number of computer literate technologists who are capable of applying IT in their specialist knowledge domains. Details are given in Chapter 8 and Appendix A.
- ⋆ Increasing the rate of technology transfer and feedback between users and suppliers; this will be the task of the BASSMATT organisation (para 9.19).

Marketing

7.14 The marketing of UK software products and services is in urgent need of revitalisation. The success of some UK companies should not be deprecated, but such success often is related to exploitation of limited home market niches. Given the high technical competence and strong historical position of UK software suppliers, the present performance is disappointing. The increasing penetration of the UK home market by foreign suppliers is reason for extreme disquiet now that software is a vital part of the industrial fabric of the UK.

7.15 We have observed several problems:

- ⋆ Comparatively healthy growth of some UK software suppliers obscures the fact that, overall, suppliers are losing international and home market share.
- ⋆ The level of commitment and investment in marketing by UK companies falls far short of that of competitors.
- ⋆ The UK has a poor image internationally.

7.16 UK software suppliers have not been notably successful in competitive international markets. This is because investment required to launch products is considerable, interaction with customers is difficult, costs of sales are high, competition is fierce and financial return results only from large sales volumes. As software becomes more of a commodity, both the cost and importance of marketing will increase.

7.17 The problems of marketing software cannot be solved easily but the rewards are high if they can be. The initiative must come from the software suppliers themselves but Government must show moral and financial commitment, as well. We believe that certain measures will be effective.

7.18 **We recommend** that UK software suppliers commit more effort and investment to publicity; this can be achieved by publications, public evaluations

and participation in trade exhibitions; Government support is recommended for these activities.

7.19 **We recommend** that UK suppliers should collaborate in gathering market intelligence; the BASSMATT initiative (para 9.19) will be tasked with assisting this activity.

7.20 **We recommend** that UK companies should enter into partnerships with indigenous companies to penetrate foreign markets and to establish the necessary user-supplier relationships; Government support should be extended to such ventures.

7.21 **We recommend** that a sustained and substantial UK software image enhancement programme be launched at home and abroad; part of this should be the establishment of a 'kite-mark' which indicates conformance to a high level of excellence (Appendix C); BASSMATT will be given the task of co-ordinating these activities.

Main government recommendations

TYPSSEA: Improving the public sector software pipeline

7.22 **We recommend** that government policies and actions on the promotion and development of software and its applications through awareness, education and training, sponsorship of industry, public purchasing, publicly funded research and development, legislation, regulations and standards should be co-ordinated by an inter-departmental, long-term strategy. This strategy is provisionally named TYPSSEA – the Ten Year Pipeline Strategy for Software Engineering and Applications.

7.23 TYPSSEA should have four main elements:

a. public purchasing to exercise demand-side leadership;
b. technology transfer acceleration;
c. public sector R&D continuity;
d. public sector in-service training.

7.24 TYPSSEA should be seen as a way of making the departmental inter-dependencies more visible, and of enabling all departments to see their roles in the public pipeline (para 6.1) as well as their own specific responsibilities.

7.25 TYPSSEA should be used as a way of representing a coherent government contribution to the STARTING mechanism.

Public purchasing and procurement

7.26 **We recommend** that all government departments use their purchasing power to exercise demand-side leadership. This should deliberately pull through new ideas already in the public pipeline (para 6.1) and should help achieve the STARTING targets for the development and application of software.

7.27 **We recommend** that the principle of lowest cost bid for software development and purchase be replaced by a more flexible policy incorporating the following objectives:

* whole life cycle costing instead of lowest construction cost bid;
* stimulus for product innovation;
* encouragement for the first use of new methods, tools and skills;
* export orientation;
* encouragement for the construction and use of re-usable software components.

Technology transfer

7.28 Technology transfer is the vital stage in the public pipeline where new ideas sponsored by publicly funded R&D need to be helped across into the industrially funded product development and exploitation stages. The UK has created for itself something of a development gap which impedes the exploitation of good UK research. Further, both the industrial and government sectors have no obvious single place to which they can look for advice on emerging software technology. In contrast to the USA's SEI, MCC and SPC (para 4.44) and Japan's ICOT the UK has no single organisation whose remit is technology transfer and which possesses the resources to do this vital job. The pace of technology transfer would be accelerated.

7.29 **We recommend** that Government set up a new technical organisation, provisionally named BASSMATT, which is to be a technology transfer organisation, similar in concept to the USA's Software Engineering Institute (para 4.44). BASSMATT should be a physical concentration of technical skills to give a national focus for software engineering and applications technology transfer. It should also provide DTI, MOD and other departments with the much needed technical muscle to implement TYPSSEA.

7.30 BASSMATT stands for British Advisory Service for Software Marketing, Applications, Training and Technology transfer.

Public sector research and development

7.31 The majority of software engineering research is public sector funded. R&D is an essential part of the public pipeline; it is the source of new ideas and, vitally, it is the main source of intelligence about overseas research.

7.32 Individual departmental budgets are too small to compete with the massive funding of America and Japan. Only by inter-departmental co-operation and informed selectivity can UK research and development keep pace with our international rivals.

7.33 The need for long-term continuity in research and development is recognised by our international competitors. The UK has a history of 'stop/go' funding which causes serious losses of efficiency, ideas, morale and highly skilled people.

7.34 **We recommend** that, through the TYPSSEA plan, co-ordination of research funding should:

- ⋆ prevent discontinuities in funding, thus giving researchers enough lead time to plan properly for expansion, contraction or direction changes;
- ⋆ quickly answer the question 'what happens after Alvey?' to avoid another discontinuity;
- ⋆ increase collaboration on R&D between industry, government and academia.

Training

7.35 The public pipeline requires skilled manpower at all levels and in all roles if it is to function efficiently. The IT skills shortage in industry has drained the public sector of most of its skilled IT manpower. This lack of skill seriously affects the quality of work, advice and decision making in the public sector today.

7.36 **We recommend** that the public sector implements a programme of in-service training similar to that recommended for industry and with similar targets.

Concluding remarks

7.37 The manufacturing and service industries face stiff international competition. The wider use, application and development of software will enhance their competitiveness. The software supply industry, too, faces competition from large

and dynamic overseas companies who are looking to dominate the UK and world markets. The continued high rate of growth in this market means that the 'window of opportunity' is still open for UK companies to grow further in world terms.

7.38 We believe that the prompt and vigorous implementation of our recommendations should give the UK a dynamic and efficient pipeline of software engineering and applications, allowing the UK's highly innovative technology 'push', to be 'pulled through' to win increased market shares by an enlightened demand-side leadership, facilitated by the skills acquired through in-service training.

8. Detailed recommendations to industry

In-service training (users, appliers, suppliers)

8.1 The **primary industrial recommendation** of this entire report is that industry initiates urgent action to update and enhance the skills of its existing workforce. This remedial action should be followed by a continuous programme of career-long, in-service training for all grades and types of staff.

8.2 The key to national success in both using and developing software is the effective education of those working with it. There is, at present, a low relevant educational attainment of most professionals involved in software. This problem has been recognised, and measures have been taken to enable the UK educational establishment, in the long term, to meet the increasing needs of industry for staff trained in software technology.

8.3 UK industry cannot afford to wait complacently for these recent changes in education to take effect. Industry must take responsibility for meeting its immediate needs. At present, UK industry makes insufficient investment in training those staff involved with software. Without the increased competence, effectiveness and productivity which training can provide, UK industry will find itself increasingly unable to deliver and deploy the software which it needs to compete.

8.4 **We recommend** that all sectors of industry using software increase their provision for in-service training to meet the following annual targets:

- ⋆ Directors, executives and senior managers should receive one week of technical training;
- ⋆ Managers should receive two weeks of technical training;
- ⋆ Technical IT staff should receive three weeks of technical training; and one week of business and marketing training.

Further discussion about in-service education and training is contained in Appendix A.

8.5 There are significant benefits to be gained from following this recommendation:

- Many managers do not appreciate the importance of software to their businesses or understand the fundamentals of software; in-service training is a means of addressing this problem.
- Many programmers do not have an appreciation of the fundamentals of business, marketing or the importance of software to industry; in-service training can correct this.
- The rapid obsolescence of technical and managerial staff, even those properly educated at one time, is unavoidable in a dynamic field such as software; in-service training is a means of addressing this problem.
- Many programmers and managers do not have an adequate mathematical and logical background; unless this is remedied they will be unable to capitalise on the emerging software engineering techniques which are mathematically based. Such mathematical groundwork will be relevant for the rest of their careers and thus will bring a continuous, long-term benefit.
- The UK has a shortage of software skills at all levels; in-service training is a means for industry to 'reskill' existing employees to meet this need.
- The scale of training requirements which the recommendations imply provide an opportunity to invest in producing cost-effective training packages and delivery mechanisms.

8.6 **We recommend** that the software producers in industry co-operate with the DTI's, STARTS software tools initiative and the STARTS Public Purchasing Group to exploit the benefits of in-service training by using STARTS/PPG backed contracts to install and try out new tools and skills; and to report on and share the knowledge about their success or failure.

Use of software (users, appliers, suppliers)

Increased rate of application

8.7 In contrast to the technically competent base of indigenous software suppliers, much of UK industry has been slow to apply software to its business activities. Consequently, industry has lost considerable opportunity to enhance its competitiveness. As long as UK software efforts are led by suppliers, industry cannot realise the full benefits of IT. Industry must take steps to ensure that it is able to take advantage of software technology by applying it appropriately.

8.8 There is a close dependency between suppliers and users of software. Without the timely availability of suitable and cost-effective applications software, many

users will be unable to compete. Without user demand, software suppliers will not produce suitable products and, ultimately, will not prosper. It is unfortunate that UK software suppliers do not have effective collaboration with an indigenous, world ranked computer hardware industry. This means that they are not in the best position to offer new and timely software products based on new hardware. UK users are unable to capitalise on new hardware developments unless they turn to foreign software suppliers. Consequently, UK suppliers are in an inherently unfavourable competitive position. Because of this, application of software in specialist domains is particularly important to the health of UK software suppliers.

8.9 **We recommend** that the application of software by industry be accelerated. We believe that the following measures will be effective:

* In-service training will greatly assist the circulation of ideas and help create a dynamic climate of software application; a major goal of the training programme should be a substantial increase in the number of computer literate technologists who are capable of applying IT in their specialist knowledge domains.
* Increasing the rate of technology transfer and feedback between users and suppliers; this will be the task of the BASSMATT organisation (para 9.20).

Process innovation

8.10 The manufacturing and service industries need to apply IT more to:

* the products they produce
* the services they offer
* the processes by which they design and produce goods and services.

8.11 Process innovation covers a huge range from automated banking tellers, office automation, computer aided design and manufacture through to computer aided navigation for motor vehicles.

8.12 Technology transfer from the software research and development community into the applier/user community is of key importance in aiding the manufacturing and service industries to compete with nations such as America and Japan who have dynamic IT industries operating symbiotically with their other industries.

Product innovation

8.13 The world market for software products and packages is the most rapidly growing sector of the software market. Only the income streams from ongoing sales will generate enough revenue to finance the R&D and capital investment needed to

stay competitive in the world market. Thus we suggest that much more effort needs to be directed towards building software products and packages.

8.14 The best source of innovative ideas for new software products is to be found amongst the users and appliers in the manufacturing and service industries because they are best placed to recognise a generic problem capable of software solution.

8.15 Suppliers and appliers/users should co-operate much more. Those software services companies who have built up expertise in particular applications areas are well placed to move into the products market and should do so. Bespoke software should be built with reusability, product spin-off and export potential in mind.

Exporting

8.16 All software should be built with the possibility of export sales in mind. Designers can, for little increase in cost or complexity, make tailoring for overseas markets much easier by for example:

* parameter driven user interface dialogues so that the natural language of the user can be easily switched. (Don't build English error messages into the code!)
* parameter driven methods for such things as units, for example imperial and metric weights and measures.
* parameter driven handling of currencies so that packages may operate with any designated local and foreign currencies.
* parameter driven standards, for example design rule checker should be able to switch between BSI/DIN/ANSI/ISO versions of requirements if overseas market requirements are country specific, for example electrical regulations.

Several of the above suggestions are related to the human-computer interface. Much greater attention needs to be paid to producing software products with better end user interfaces than present offerings. Interfaces which are tailorable to different nationalities are one example. End users have differing levels of computer literacy, differing levels of application domain skills, differing educational and cultural backgrounds. Careful consideration of the end users' problems, requirements and skills will help software designers to produce more usable, and hence more successful, products.

Investment

8.17 The user/applier communities in manufacturing and service industries must increase their investment in software to stay competitive. This capital investment will only give a good return if it is accompanied by a proper level of in-service

training and tools investment. Software is not a magic black box technology. It embodies the competitive edge for many activities and so requires continuous update and improvement.

8.18 The software appliers and suppliers will, in the next ten years, see software engineering move from being labour-intensive in the 1970s to being very capital-intensive from 1990 onwards. Capital investment will be required to raise the amount of computer power per software engineer significantly. The UK is already behind the USA in this respect. Capital investment will be needed for the new software tools which are becoming available. The larger overseas suppliers are building Integrated Project Support Environments and Information Systems Factories (**5**) facilities; these are large investments.

8.19 Stocks of reusable software components must be built and maintained; this will require investment. The development of products from these components will require significant investment as will, most of all, the marketing of these products. Again, such investments will only be profitable if accompanied by a continuous programme of in-service education and training.

Report to shareholders

8.20 **We recommend** that all user, applier and supplier companies indicate, in their annual reports to shareholders, the progress they have made in training their workforces in IT skills; the investment made in improving their production processes via IT; the new products they have brought to market which exploit IT; and the overall investment made in software and IT.

Marketing (appliers, suppliers)

8.21 The marketing of UK software products and services urgently needs to be revitalised. While the success of some UK companies should not be depreciated, such success is often related to the exploitation of a small market niche or sector. There are many such market opportunities, particularly in the international market-place. In general, the UK software industry is failing to take advantage of them.

8.22 The following problems need to be addressed in order to achieve the goals set out in our main recommendations:

- ⋆ An apparently healthy growth rate in individual companies obscures the fact that the UK software industry is losing market share, both at home and abroad.
- ⋆ Internationally, UK software products and services suffer from lack of image.

International customers do not perceive any advantage to 'buying British'.

- UK software products and services do not satisfy the needs of the international market-place.
- The up-front investment required to introduce a new product into overseas markets is substantial. The costs of sales are significantly higher in the key USA and Japanese market-places than at home. Competition is fiercer, and return on investment results only from sales volume and follow-on business.

8.23 The problems for the software services industry are particularly acute, resulting from the obvious difficulties of interacting directly with overseas customers. This sector will need to become more strongly based on packages and reusable components.

8.24 We believe that the level of investment in marketing by UK companies falls far short of that of our competitors, and that the industry suffers accordingly. The following table, taken from a recent report by the US Department of Commerce (2), shows how USA personal computer software companies allocate their return on revenues:

Marketing	35%
Management	20%
Production	15%
R&D	15%
Profit	15%

8.25 UK software companies need to take a much more positive attitude to marketing, particularly in the international market-place. The initiative must come from the individual companies, but Government must show commitment to back this up with the support identified in our various recommendations.

8.26 **We recommend** that a sustained and substantial UK software image enhancement programme be launched at home and abroad; part of this should be the establishment of a 'kite mark' which indicates conformance to a high level of excellence (Appendix C); BASSMATT should be given the task of co-ordinating these activities (para 9.20).

8.27 **We recommend** that more effort should be put into the support provided by UK embassies. The commercial and scientific attaches should be fully and regularly briefed, and should be made very aware of the new value and significance that the UK places on its software industry.

8.28 **We recommend** that UK software companies commit much more effort into creating product awareness. This can be achieved by placing technical and

marketing articles in the international trade journals, by ensuring that third party reviews and evaluations are carried out and published, and by participating in international trade shows and exhibitions. UK software must be seen to stand up well to international competition.

8.29 **We recommend** that UK software companies should collaborate with each other in gathering market intelligence. BASSMATT (para 9.20) should help with this task.

8.30 **We recommend** that UK companies should enter into partnerships with indigenous companies to penetrate foreign markets and to establish the necessary user-supplier relationships; Government support should be extended to such ventures.

8.31 Special effort needs to be put into identifying new market areas. Government incentives should be provided to encourage the development of products in these areas. Such product development should be based on proven R&D, which is the subject of separate funding. The task is to bring a substantial idea (often embodied as a computer program) to the level of maturity and packaging so that it becomes an effective and easy to use tool in the hands of the customer. It must also improve the customer's productivity in some way, and be competitive. Areas for the future include value added networks, man/machine interaction, the electronic office, process control software, integrated CAD/CAM systems, and on-line databases. Products which take a systems approach and exploit the new technology should be particularly encouraged. EUREKA and the DTI 'commercial demonstrator' (para 9.17) scheme should provide valuable stimuli to this objective.

8.32 It is our opinion that a disproportionate amount of effort is being applied to bespoke software, to the detriment of software products. The dependency on bespoke government contracts can only make matters worse. The software industry needs to take a speculative approach by developing families of compatible modules, which can be assembled into products, or, with little extra effort, be tailored to fit bespoke requirements. Marketing has a significant role in the specification of such families and the de facto international standards they must meet.

Research and development (appliers, suppliers)

8.33 In our opinion industry must do much more short- and medium-term R&D. This increased activity, together with in-service training, will give a more vigorous and expert base from which to develop products, and to import new long-term work from academia, thus greatly reducing the technology transfer gap.

8.34 Medium-term work should be carried out collaboratively with academia and with other UK and European industrial partners to share the costs and make best use of limited, high calibre talent.

Restructuring (suppliers)

8.35 The UK supply industry is showing typical UK insularity in the way it remains fragmented. Small companies, though growing, are expanding less rapidly than the world market and are not generating the revenue to finance the right level of R&D and marketing.

8.36 Such companies show a marked reluctance to merge into larger units, be absorbed into larger industrial groups or to collaborate in attacking export markets. So divided, they will be conquered. The UK software suppliers must be prepared to co-operate more both between themselves and with the user/applier industrial sectors.

8.37 Most UK software houses are too small to tackle world markets. If they continue to compete with each other for the home market then we foresee that the large American companies will either take them over or just take over their market share until only the MOD market is available to the surviving UK companies.

8.38 The UK suppliers must restructure into larger internationally competitive units. We are not in favour of direct government action to achieve this; however, if this restructuring does not occur naturally, and soon, then our pessimism will likely be justified.

8.39 We suggest that the industry should see itself as the UK component of the European struggle against American and Japanese competition.

9 Detailed recommendations to government

The need for inter-departmental co-operation and co-ordination

9.1 Previous chapters referred to the multiplicity of ways in which government affects the development and application of software, software engineering and IT generally. These include:

- national policy
- national targets and expectations for IT development
- national awareness and attitudes
- education and training
- public sector research and development
- public sector purchasing and procurement
- sponsorship and support of industry
- regulations and standards, national and international
- the legal framework for the applications of IT.

9.2 Responsibility for these activities is split between many government departments, including DTI, MOD, DES, DHSS, Home Office and CCTA. This spread of responsibilities for the advancement of developments and applications of IT does not seem to us to provide a coherent framework for policy making for such a nationally important subject. Decisions on one aspect of IT may have repercussions in a very different area. We believe that it is necessary that all factors which will influence the development and application of IT in the national interest should be studied together, and that departments' plans and actions are co-ordinated and monitored.

9.3 There is also the difficulty that costs and benefits from the introduction of IT systems can fall to different government departments. Separate financial targets and accounting systems do not easily allow the redistribution of costs and benefits. Projects where the net national benefit would be substantial may not then come to fruition. There is need for a system of project cost and benefit distribution which can overcome departmentalism.

9.4 We believe that a focal point is necessary, in view of the number of organisations involved, to improve awareness in government, to promote a programme covering both projects and publicity, to improve internal communications and provide necessary couplings, to avoid delays, and to create a positive public consciousness of IT. There is a need to make existing UK efforts more coherent.

TYPSSEA: Improving the public sector software pipeline (all departments)

9.5 **We recommend** that government policies and actions on the promotion and development of software and its applications through awareness, education and training, sponsorship of industry, public purchasing, publicly funded research and development, legislation, regulations and standards should be co-ordinated by an inter-departmental, long-term strategy. This strategy is provisionally named TYPSSEA – the Ten Year Pipeline Strategy for Software Engineering and Applications.

9.6 TYPSSEA should have four main elements:

a. Public purchasing to exercise demand-side leadership;
b. Technology transfer acceleration;
c. Public sector R&D continuity;
d. Public sector in-service training.

9.7 These four elements are the major stages in the public 'research-to-customer' pipeline (para 6.1). TYPSSEA should be a long-term strategy in which the departments associated with the pipeline form a series of joint targets and plans to increase the efficiency of the pipeline. Co-operation should occur through the use of public purchasing power to pull through new applications and development methods into commercial use; through the proposed technology transfer organisation to accelerate the pull through to commercial demonstration of new methods, tools and applications and push them towards commercial use; and collaboration on R&D planning to eliminate the discontinuities in R&D funding, whose 'stop/go' effect seriously damages the vital work of generating the staff and ideas which form the pipeline's feedstocks.

9.8 We suggest that the DTI take the lead in bringing TYPSSEA into operation as DTI has overall responsibility for the health of the manufacturing and service industries, and in particular the software and IT industries, that is, most of the industries involved in the pipeline. Also, as a member of the Alvey Programme, DTI is well placed to help liaise with MOD, DES, industry and academia in the research and development areas.

9.9 TYPSSEA should be seen as a way of making departmental interdependencies more visible, and of enabling all departments to see their roles in the public pipeline as well as their own specific responsibilities.

9.10 TYPSSEA should be used as a way of representing a coherent government contribution to the STARTING mechanism.

9.11 TYPSSEA should be used by purchasers to plan their major projects with a better knowledge of the emerging technology; it should allow the R&D agencies the opportunity to recommend new innovations to the purchasers; it should allow purchasers to influence the R&D programmes so that relevant applications problems are tackled; it should allow various departments to improve long-term manpower planning; it should allow the influence on procurement of education and training policies and curricula to be recognised (as when new recruits move up the management hierarchy to influence purchasing policy).

Public purchasing and procurement (all departments)

9.12 **We recommend** that all government departments use their purchasing power to exercise demand-side leadership. This should deliberately pull through new ideas already in the public pipeline (para 6.1) and should help achieve the STARTING targets for the development and application of software.

9.13 **We recommend** that the principle of lowest-cost bid for software development and purchase be replaced by a more flexible policy incorporating the following objectives:

- ⋆ *Whole life cycle costing.* Software projects, especially large ones, take several years to develop a product which then stays in service for 10-20 years but undergoes continuous modification. For software the in-service maintenance costs can be much greater than the initial construction costs if the initial design is done on a lowest-cost basis. An increase in effort at the design stage can have massive savings over the whole life cycle. The purchasing departments and the Treasury should therefore adopt life cycle costing for software. This will result in better 'value for money' overall.
- ⋆ *Stimulus for product innovation.* Purchasing departments should look beyond their own immediate requirements to see if their specifications can be used to pull through new ideas in the public pipeline which have a broader market than just the purchasing department.
- ⋆ *Encouragement for the first use of new methods, tools and skills.* At some small additional cost the purchasing departments can pull through the emerging new techniques of software engineering and demonstrate their effectiveness. If such

new techniques are encouraged by the purchaser then industry will be encouraged to invest in the new tools and training.

- ★ *Export orientation.* Purchasing departments should ensure that their specifications are not parochial to the UK, thereby eliminating the export opportunity. Specifications should be explicitly formulated to deliberately encounter export, for example multiple currency facilities (£,$, DM) ISO not just BSI standards; multiple natural language version potential for NATO and European use; ADA not CORAL (see para 8.16 'exporting').
- ★ *Encouragement for the construction and use of re-usable components.* Purchasing departments should encourage the construction of software which can be re-used on later projects. This will require some additional cost on any specific project but over time will lead to the spin-off creation of a national asset base of re-usable components. Industry can then exploit these re-usable components to reduce the cost of subsequent public contracts and win new non-public contracts. Re-use is an important software engineering technique which requires demand-side leadership to pull it through into commercial practice.

9.14 **We recommend** that all government departments purchasing software should join the existing STARTS Public Purchasers Group. The STARTS/PPG should expand its term of reference to include all types of publicly purchased software. The STARTS/PPG should receive additional resources to improve on the good work already achieved. The STARTS/PPG should lead the drive towards public purchasing specifications for software projects which encourage:

- ★ whole life cycle costing
- ★ public pipeline pull through of new product ideas and process innovation
- ★ export orientation
- ★ re-usable component construction
- ★ first use of new tools and techniques
- ★ new in-service training courses to improve skill levels.

9.15 **We recommend** that the STARTS/PPG should work to improve the dialogue between customers (users and appliers), suppliers and researchers. This should improve the effectiveness of demand-side leadership by giving early notification of public requirements, expanding on the long-term targets set by STARTING. It should also improve the demand-side's knowledge of what is in the research and supply-side pipeline – the technology push component. A general improvement in the flow of information between the demand-side and the supply-side should improve the rate of development of both the supply industry and the user departments.

9.16 **We recommend** that the remit and resources of the STARTS Public

Purchasers Group be expanded to allow the major private sector purchasers to join STARTS/PPG and that the STARTS/PPG's remit be widened to include all classes of software, not just 'real time' applications.

9.17 **We recommend** that public purchasing power be used to fund two types of 'demonstrator' project: the research demonstrator and the commercial demonstrator (see Figure 8). These demonstrators should increase the flow of the pipeline from research to product and help counter the UK tendency to fail to bring good ideas up to the demonstration stage.

9.18 **We recommend** that UK procurement be seen in a European context and be used to further the development of European software and IT industries.

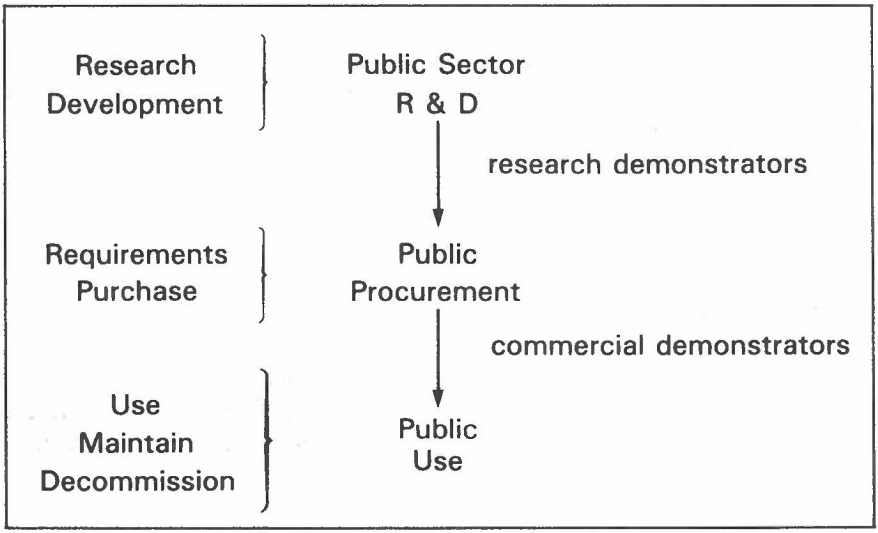


Figure 8. The public pipeline: demonstrator projects

Technology transfer (all departments)

9.19 It is much more expensive to market a product internationally than just to develop it. It is much more expensive to develop a product than to do the originating research. Technology transfer is the vital stage in the pipeline where new ideas sponsored by publicly funded R&D need to be helped across into the industrially funded product development and exploitation stages. The UK has a development gap which impedes the exploitation of good UK research. Further, both the industrial and government sectors have no obvious single place to which they can look for advice on emerging software technology. In contrast to the USA's SEI, MCC and SPC (para 4.44) and Japan's ICOT, the UK has no single organisation whose remit is technology transfer and which possesses the resources to do this vital job.

9.20 **We recommend** that government set up a new technical organisation, provisionally named BASSMATT, which is to be a technology transfer organisation, similar in concept to the USA's Software Engineering Institute (para 4.44). BASSMATT should be a physical concentration of technical skills to give a national focus for software engineering and applications technology transfer. It should also provide DTI, MOD and other departments with the much needed technical muscle to implement TYPSSEA.

BASSMATT: technology transfer acceleration

9.21 One of our major recommendations to government is the formation of a technology transfer organisation, provisionally called BASSMATT, the British Advisory Service for Software Marketing, Applications, Training and Technology transfer.

9.22 BASSMATT should provide a national focus for software engineering and its applications. It should be modelled on the USA's SEI (para 4.44) in terms of scale and quality. It should be located in 'software valley' (the M4 corridor), the heart of the software industry.

9.23 BASSMATT's functions and effort should be as follows:

* 30% Technology transfer
* 25% Independent technical advice
* 15% Marketing and legal support
* 15% Collaborative research
* 10% 'After Alvey' programme support and collocation
* 5% STARTING and TYPSSEA support

9.24 BASSMATT should speed the flow of technology transfer between the user/applier/supplier communities of manufacturing and service industries, the IT and software industries, government departments and academia. The following activities are suggested:

- ⋆ Disseminate information via reports, seminars, newsletters, electronic databases, electronic-mail, bulletin boards etc.
- ⋆ Actively demonstrate state-of-the-art tools and products (the 'showcase environment'). BASSMATT should have an in-house Integrated Project Support Environment (IPSE) acting as a Software Production Centre (see below) and a Software Re-usable Components Brokerage (see below) as suggested in the Alvey Report (**5**).
- ⋆ The Software Production Centre (SPC) should not be a research project but the latest IPSE technology functioning as a working software factory. This will enable organisations to try out new techniques 'for real' in order to make a better selection and case for their own in-house investment. The SPC will be a 'showcase' of the best software engineering methods and tools.
- ⋆ The Software Re-usable Components Brokerage (SRCB) should be a database holding requirements, specifications, code, test data, documentation etc. Components should be collected from publicly funded projects and be made available to other such projects, thereby exploiting the benefits of the public purchasing recommendations encouraging the funding of re-usable components (para 9.13). The SRCB's facilities should be made available via a communications network (this will be a driver to resolve various important issues such as security and licensing for such a new distribution technique).
- ⋆ The SPC/SRCB/IPSE will be located at the 'software valley' site. The newly established software tools demonstration centre at NCC, Manchester, should become an outstation of BASSMATT. The Software Tools Demonstration Centre could be expanded to offer the same SPC/SRCB facility in Manchester. Other such outstations should be considered if a critical mass of companies can be identified to be served who are remote from the 'M4 corridor' but centred around a potential outstation.
- ⋆ BASSMATT should be in touch with all the best UK R&D work to 'pull through' results.
- ⋆ BASSMATT should seek out the best overseas work and transfer information about it to the UK community.
- ⋆ BASSMATT should support the in-service training initiative by:
  - developing course material including distance learning material
  - tutorial use of the Software Production Centre
  - training the trainers

- hosting short courses
- investigating the quality of educational courses
- investigating the efficacy of professional certification.

Independent technical advice (25%)

9.25 BASSMATT should provide a focal point to which government departments, and industry too, can turn for impartial expert advice. BASSMATT's activities should include:

* evaluation and cost benefit studies of new software engineering methods, tools and products including helping STARTS/PPG tools initiative;
* the development and assessment of new standards (Appendix C);
* an independent testing service to evaluate methods, tools and products. This should include work towards quality certification (Appendix C);
* contribute to the work on safety critical aspects of software engineering (Appendix B);
* act as a source of independent technical expertise for the STARTS/PPG public procurers;
* offer general advice to the user/applier/supplier communities.

9.26 BASSMATT should build up expertise in the formulation and formalisation of software standards and designs. BASSMATT staff should participate in the work of standards bodies. A source of expertise in the application of formal methods for software engineering is going to be needed by both government and industry in the next few years, as software technology increases its mathematical basis. The need for better standards is discussed in Appendix C.

Marketing and legal support (15%)

9.27 BASSMATT should help identify world-wide market opportunities arising from technical innovation, and give feedback from government users to suppliers. It should help disseminate information from market research studies (para 7.4). It should also help the DTI's legal side with technical advice relating to issues such as piracy, safety, warranties etc.

Home to the 'after Alvey' research programme (10%)

9.28 The management of the research programme taking up the Alvey role (as recommended in para 9.42) should be physically collocated with BASSMATT. This will ensure maximum pull-through from the research programme into the recipient industries by allowing excellent liaison opportunities. The research programme

management will benefit from the technical environment and support which BASSMATT will be able to offer. Such collocation of research programme management and technology transfer organisation should provide a natural focus for orchestrating the UK participation in European-scale programmes such as EUREKA and ESPRIT.

9.29 BASSMATT will also make a natural home for some research demonstrators (para 9.17).

Research (15%)

9.30 A small internal research programme should be undertaken in collaboration with industry and academia. This will give BASSMATT people who are 'plugged in' to the research grapevine; these are the right people to pull innovations into BASSMATT. Research is also a vital career development mechanism, enabling staff to be rotated between research and technology transfer activities as part of their in-service training. The desirability of having, as a BASSMATT component, a research 'centre of excellence' should be investigated.

STARTING support (2%)

9.31 BASSMATT should support STARTING by providing the technical effort needed to produce the annual report, the annual conference organisation and the organisation of the presentation to STARTING of evidence from industry, academia and government.

TYPSSEA support (3%)

9.32 BASSMATT's primary function on behalf of TYPSSEA should be to speed up the flow of technology along the public pipeline, but BASSMATT should also support TYPSSEA by providing technical expertise and a common point of inter-departmental contact to help formulate objectives, strategy and tactics.

Organisation

9.33 BASSMATT should:

- have an initial 15 year lifetime with five-yearly reviews;
- be of similar size and quality to the USA's SEI or Japan's Sigma project (para 4.44);
- be located in 'software valley' between London and Bristol;

- have both permanent staff and secondees from government, industry and academia;
- subcontract some of its work where appropriate;
- as a national showcase, be staffed and equipped as well as rival national institutes such as SEI. The right quality of staff will not be easily recruited but a principle of 'top quality' should be initially established and maintained. If the quality of BASSMATT is not of world class then its next five-yearly review should recommend its closure.

Public sector research and development (all departments)

9.34 The majority of software engineering research is public sector funded. R&D is an essential part of the public pipeline; it is the source of new ideas and, vitally, it is the main source of intelligence about overseas research.

9.35 Individual departmental budgets are too small to compete with the massive funding of America and Japan. Only by inter-departmental co-operation, and informed selectivity, can UK research and development keep pace with our international rivals.

9.36 The UK R&D effort should be viewed as providing the feedstock for the UK contribution to the European community in which the major collaborative programmes should be executed.

9.37 The industrial, academic and inter-departmental co-operation exhibited by the Alvey Programme is an encouraging sign that the UK software community can work collaboratively at both technical and organisational activities.

9.38 **We recommend** that TYPSSEA be used to co-ordinate the research and development programmes of all departments to ensure that serious discontinuities do not occur as a result of either starting new programmes (which take time to become effective) or stopping existing programmes without adequately planning what should replace them (or not) so that enough time exists for the R&D community to cope smoothly with the change.

9.39 **We recommend** that the Alvey style collaboration on software R&D issues be continued and expanded to include those agencies and departments (such as UGC, DHSS) who are not currently involved.

9.40 **We recommend** that the 'what happens after Alvey?' question be resolved as soon as possible.

9.41 The UK must avoid a 'stop/go' approach to R&D. This causes loss of morale, loss of irreplaceable staff to overseas competitors, breakdown of collaborative working relationships, rundown of infrastructure and a large loss of momentum. With the ever increasing pace of international competition a smooth, continuous transition from the current Alvey programme to its successor is vital. The plan for such a successor is needed urgently.

9.42 We make the following suggestions for the organisation of a successor to the Alvey Programme:

* It must be collaborative between government, industry and academia.
* It must have its own budget, derived from government departments, but under its own control.
* It must have the ability to fund at a flexible level, say from 10-100% depending on the size of the company, the size of the project and the balance of research content versus product development. This could be achieved within an overall, bottom-line budgetary constraint of a 50:50 split between government and industrial contribution.
* It should have the remit to fund research and commercial demonstrator projects (para 9.17).
* It should fund product innovation research as well as 'enabling' technology.
* It should be able to fund the development of advanced training courses and materials to speed the dissemination of new ideas.
* Its HQ should be collocated with the BASSMATT to ensure adequate technical support and close co-operation between R&D and technology transfer.

9.43 **We recommend** that, in overall terms, industry should increase the level of its short and medium-term R&D so that the academic community can restore the balance of its work in favour of the longer-term work while maintaining the industrial collaboration links built up by the Alvey programme.

9.44 The above set of recommendations should give the UK a balanced and efficient pipeline of software engineering and applications, with the highly innovative technology push, at which the UK is good, being pulled through by an enlightened demand-side leadership, facilitated by the new skills acquired through the in-service training programme.

Training (all departments)

9.45 The public pipeline requires skilled manpower to all levels and in all roles if it is to function efficiently. The IT skills shortage in industry has drained the public

sector of much of its skilled IT manpower. This lack of skill seriously affects the quality of work, advice and decision-making in the public sector today.

9.46 **We recommend** that the public sector implements a programme of in-service training similar to that recommended to industry and with similar targets.

DTI recommendations

TYPSSEA

9.47 **We recommend** that, as DTI has overall responsibility for the health of the manufacturing and service industries, and in particular for the software and IT industries, DTI should be responsible for leading the formulation of TYPSSEA.

Demand-side leadership

9.48 **We recommend** that DTI help organise the increase in demand-side leadership called for in TYPSSEA. We suggest that this should be done by expanding the remit and resources of the STARTS Public Purchasing Group.

9.49 We suggest that DTI examines the potential benefits of inviting the major private sector purchasers to join the STARTS/PPG and that its remit be widened to include all major software purchases, not just 'real-time' applications.

Technology transfer

9.50 **We recommend** that DTI mounts a major exercise to improve the transfer of technology along the public pipeline and between the user/applier/supplier communities. This exercise should encompass the following:

- ⋆ support for the in-service training initiative in industry;
- ⋆ pull through of the research results of public sector R&D programmes, specifically Alvey and ESPRIT, including government support for research demonstrators (para 9.17);
- ⋆ support for commercial demonstrator (para 9.17) projects of innovative software engineering technology in real user applications to encourage take-up more widely;
- ⋆ support for the creation of the technology transfer organisation, BASSMATT (para 9.20);
- ⋆ increased support for the STARTS/PPG software tools initiative;
- ⋆ continue support for 'awareness' programmes;

* support cost-benefit studies of new methods, skills and tools.

Support for product innovation, marketing and exporting

9.51 **We recommend** that the Software Products Scheme and Support For Innovation Scheme be extended and enhanced to stimulate the software products industry and encourage user/applier companies to spin-off their bespoke developments into products.

9.52 We applaud the marketing and exporting aspects of the Market Entry Guarantee Scheme and Software Products Scheme and suggest that these mechanisms be enhanced to stimulate greater marketing efforts from industry, as marketing is the key to reaching the STARTING targets.

9.53 We suggest that DTI explore the idea of the Computer Services Association and National Computing Centre organising regular export marketing seminars to identify software export opportunities including the possibility of collaboration in overseas markets.

9.54 DTI should help UK firms participate in European initiatives such as EUREKA.

Further study

9.55 **We recommend** that DTI conducts further studies of four important issues raised in the Working Group. We feel these issues require action within the next five years:

a. the legal and technical issues surrounding safety-critical software, that is, software products whose (faulty) operation could endanger public safety;
b. the potential of software product quality certification (the 'software kitemark');
c. the role of new technical standards as a means of helping the UK industry tackle the world market;
d. the role for the certification of practising software engineers.

These subjects are discussed further in Appendices A, B and C.

Fiscal and legal recommendations

9.56 **We recommend** that DTI examine the scope for fiscal measures, for example tax incentives, to support the main recommendations, especially in-service training and investment in software tools.

9.57 **We recommend** that DTI examine what, if anything, needs to be done about copyright laws and software piracy; about the protection of Intellectual Property Rights to software innovations; and about advising exporters on legal problems associated with selling software abroad.

9.58 **We recommend** that DTI examines the legal issues surrounding warranties and guarantees, disclaimers and the Trades Description and Health and Safety Acts in relation to software suppliers, their products and customers.

9.59 **We recommend** that DTI examines the problems facing UK companies arising from import/export controls such as COCOM to see what can be done to reduce the delays and frustrations which have been reported.

Schools

9.66 **We recommend** that schools consider the following suggestions. Touch typing and 'mouse' skills should be taught as basic skills. The basic mathematics relevant to understanding IT should be taught. We suggest that Computer Studies for school children under 16 should be based on sound problem analysis and structured program development. The teaching of BASIC should be phased out as more powerful systems become available. A common core syllabus for Computer Science at 'A' level should be constructed. This should include a good groundwork in the mathematical aspects of software engineering, as well as its practical application. The 'A' level should be a suitable qualification for entry to a degree course in Computer Science.

MOD Recommendations

Procurement

9.67 **We recommend** that MOD strongly supports the STARTING and TYPSSEA policies and uses its vast influence and purchasing power to improve the quality of demand-side leadership and its goals of technology pull-through, export spin-offs etc. We welcome MOD's move to strive for more commercial spin-off. We look to MOD to use its procurement executive to encourage suppliers to work to the best software engineering standards and thus for MOD to strongly support the in-service education and training initiative.

Ada

9.68 **We recommend** that MOD takes strong and immediate steps to improve current UK Ada technology.

An Acard report.

Software: A vital key to UK competitiveness

ISBN 0 11 630829 X

Addendum: please note that the following paragraphs have been omitted from page 68.

DES recommendations

Research

9.60 **We recommend** that the DES and its agencies recognise that software engineering is an experimental engineering discipline requiring considerable equipment and technician resources. Planners and committees should recognise that software engineering is capital intensive. Teaching engineers and scientists how to apply software requires facilities comparable to industrial quality and sophistication. Rather than just acquiring the ability to create small programs 'from scratch', engineers and scientists today need to know how to solve applications problems by assembling and adapting software components, and how to maintain the solutions. This can only be done with facilities realistically comparable to best industrial practice. Software Engineering teaching and research require facilities of greater sophistication than current best industrial practice.

9.61 **We recommend** that SERC recognise the ever-increasing national importance of IT by elevating its Information Engineering Committee to be a full Board. SERC should continue its good work encouraging the collaboration between industry and academia, whilst maintaining a careful balance between that and its long-term work in IT research.

9.62 **We recommend** that DES strongly supports an 'after Alvey' research programme.

9.63 **We recommend** that DES, through its various agencies such as UGC and SERC, supports the academic participation in European projects by funding the necessary pre-proposal work and the additional costs of European collaboration.

Standards

9.64 **We recommend** that DES, through its agencies, supports the participation of academics in IT standards research, development, formulation and promulgation. It is vital that standards include the best work on the subject (See Appendix C).

In-Service Training

9.65 **We recommend** that DES supports the industrial in-service training initiative (Appendix A). Much good work has already been done. Further help could be given by:

* encouraging the 'reuse' of teaching material by industrial trainers
* organising MSc Courses to cater for day release students, evening classes, distance learning etc, so that students may pursue their studies whilst maintaining a relatively normal working life
* encourage the training of new trainers who will be needed by industry

Cabinet Office
June 1986

London: Her Majesty's Stationery Office

Research

9.69 **We recommend** that MOD continues to support collaborative research 'after Alvey'.

Technology transfer

9.70 **We recommend** that MOD strongly supports the BASSMATT mechanism. **We also recommend** that MOD continues to improve its technology transfer efforts including demonstrator projects, support for standards development and the promulgation of the IPSE concept.

Other government departments

9.71 **We recommend** that other government departments such as DHSS strongly support the TYPSSEA and BASSMATT recommendations, in view of their significant requirement for software and their considerable purchasing power.

9.72 **We recommend** that other government departments review their interest in the safety-critical aspects of software discussed in Appendix B. The development of software-controlled, patient monitoring and care systems means that we would expect DHSS to take an interest in the safety issue.

10. Detailed recommendations to the professional institutions

10.1 The IEE initiative to boost software engineering within the Institution is welcomed and encouraged. The BCS Professional Development Scheme is also welcomed and encouraged.

10.2 **We recommend** that the IEE lead a study into the issue of professional certification of software engineers, safety critical software and quality certification as described in the appendices. The IEE should involve the Engineering Council, the BCS, industry, academia and government in these studies.

10.3 **We recommend** that the IEE and the BCS strive to encourage industry and government to implement the in-service training recommendation. (See Appendix A.)

11. References

1. *Software Engineering Economics*
   BOEHM, B. W.
   Prentice Hall, 1981

2. *A Competitive Assessment of the United States Software Industry*
   Miles T and Ahlberg R
   Office of Computers and Business Equipment
   US Dept of Commerce
   GPO Stock Number 003 009 004364
   February 1985
   (summarised in IEEE Computer, March 1985)

3. Information Technology Economic Development Council, NEDO

4. *Public Acceptance of New Technologies: New Communication Technology and the Consumer*
   Hartley et al
   PREST, University of Manchester, 1985

5. *Alvey Software Engineering Strategy*
   Alvey Directorate, DTI
   November 1983

6. *Submission to ACARD Working Group*
   Institution of Electrical Engineers, 1985

7. *Signposts For The Future*
   IT Skills Shortages Committee
   Final Report
   DTI, 1985

12. Appendix A: Education and professional qualifications

The problem

A.1 The key to UK success in Information Technology (IT) is the appropriate and effective professional education of all who engage in it – from directors and managers, through systems analysts and programmers to marketing and sales staff.

A.2 A noticeable problem in the IT industry is the low relevant educational attainment of many practitioners. Even those with appropriate qualifications have little incentive or opportunity to keep abreast of the subject. Furthermore, the effectiveness of those qualified may be inhibited by the lack of education of their colleagues and of those occupying more senior positions.

A.3 We wish to distinguish carefully between 'education' and 'training'. Education is intellectual instruction which is of general and long-term value. Most people perceive formal education as emanating mainly from schools, universities and the 'educational system'. Training is practical instruction which tends to be orientated towards specific goals, usually vocational. Most companies offer only in-service training to their employees.

A.4 At present, qualification tends to be based mainly on work experience. Low educational attainment is difficult to avoid in a discipline undergoing a high rate of development and rapid diversification of application area. In Britain this problem is compounded by several factors:

- ⋆ the absence of a tradition of post-educational training for practitioners in the IT industry;
- ⋆ high turnover of IT staff, which makes employers reluctant to invest in any but short-term training of immediate applicability;
- ⋆ a prevalent belief that education is the responsibility solely of government;
- ⋆ a strong tradition of specialisation during and after formal education;
- ⋆ failure, in the IT industry, for individuals to obtain, through their own efforts or through those of their employers, necessary regular in-service education.

* lack of common culture, experience and education in the IT industry.

A.5 Various bodies have made recommendations and taken action to increase the numbers of initially qualified entrants to professions associated with IT. We have not considered it our task to comment on the adequacy of these increases; rather, we must point out the well-known dangers of recruiting increased numbers of inexperienced engineers, when their senior colleagues and managers are not educationally qualified or competent to direct and supervise their activities.

A.6 As in other branches of engineering, it takes time to establish cadres of senior professionals with the right education, experience and personality to direct the work of others. To meet urgent needs, this process must be accelerated; and the quickest way to do this is to raise the educational level of those who already have the requisite experience and personality.

A.7 The British educational system and professional institutions have many merits and an enviable record of achievement in the established branches of engineering. It is essential that the development of engineering education in the IT industry should aim at the same high standards.

Towards a solution

A.8 The precondition for a solution is to establish the principle that every responsible professional in IT will expect to supplement his/her formal education in software engineering more than once in his/her career. IBM probably is the most successful IT company in the world. Their investment in in-service education and training is outstanding. Employees are required to undertake several weeks annual training, no matter what their position in the company. IBM has implemented a policy, laid down some five years ago, that every programmer and manager of programmers should attend between two and six weeks of in-service education in the formal methods of software engineering. It is difficult to believe that a considerable part of IBM's success is unrelated to their commitment to in-service education.

A.9 As a nation we should set ourselves a similar target. Success in IT often requires, in a single person, technical competence, applications knowledge and business skill. In addition to the central core courses in programming methodology and software engineering, we also envisage more specialist courses on such topics as the needs of business and commerce, communications and protocols, process control, man/machine interfaces, knowledge engineering, management, marketing, programming language standards, other emergent standards, and safety-

critical programming. None of these will compete with more specialised training courses in the use of proprietary hardware or software products.

A.10 Action has been taken by various groups (for example the IT Skills Shortages Committee (**6**)) to encourage the educational system to increase the numbers of qualified entrants to the IT industry. Even if the output of professionally educated IT personnel should become adequate, it will be many years before those educated have acquired sufficient experience and seniority to have an effect on the industry as a whole. Attempts to increase the quantity of those issuing from the educational system must not divert attention from the immediate need to improve the quality and effectiveness of existing practitioners.

A.11 **We recommend,** therefore, that a wide scale programme of in-service education be established by, and in, UK industry as a matter of high priority. The goals of this programme should be to improve the quality, flexibility and productivity of a significant proportion of those engaged in IT. The programme must not be confined to the insular aspects of software engineering and computer science, nor should it compete with specialised industrial training. Instead, the initiative should strive to impart appropriate familiarity with those subjects necessary for effective and professional conduct within the IT field. As such, the programme must be able to address the many levels of education required by such a diverse target population.

A.12 British professional institutions have an enviable record of success in establishing in-service education in many traditional branches of engineering. It is sensible to capitalise on the reputation and experience of the existing institutions. Success of the in-service education programme will require the co-operation of the Microelectronics Education Programme, universities, polytechnics, the Science and Engineering Research Council, the University Grants Committee and the Standing Committee for University Entry. The Government might participate both from a regulatory and an employer perspective. Above all, success will require commitment and participation from industry. The IEE and BCS have the necessary political experience to forge the necessary collaboration between the diverse organisations involved.

A.13 A prime mover in the setting of educational and professional standards for certification of engineers is the Engineering Council; in the case of software engineering, the Council works through the IEE and the BCS. Both these institutions are actively reconsidering their membership criteria for software engineers. Ideally, these bodies should collaborate on setting up or accrediting the required examinations, courses, and text-books, starting perhaps in the technically most demanding and most urgent area of safety-critical programming.

Curricula

A.14 It is unreasonable to expect that any formally approved syllabus and examination will in the foreseeable future keep pace with the rapidity of developments in Information Technology. It is therefore necessary to take more immediate steps to stimulate the development of new courses on offer from research and educational institutions, both public and commercial, for professional updating of programmers from industry and public service. By instituting an educational policy for programmers in public service, the Government could do much to stimulate demand; and the supply side can be stimulated by explicit commissions of course material, for example from the Open University and other sources capable of more rapid response.

A.15 The content of the new courses should be designed by those with the relevant practical and research experience, in close consultation with the organisations most likely to send students on the course. Each course should be given initially by those who have designed it.

A.16 Establishing curricula for an in-service educational programme of wide scope is not simple. The curricula must be accepted by all concerned as being technically robust and having long-term viability. We believe that the institutions recommended (that is, the IEE and the BCS) have expertise in establishing professional education schemes and will be able to accredit acceptable curricula. In-service education also would gain considerable prestige if it were acceptable as credit for traditional qualifications from professional and educational establishments. We recommend, therefore, that this be one of the goals for those establishing the programme. We recommend that the curricula should cover a wide range of subjects, but in no greater detail than necessary to provide a useful educational base. Some essential topics are:

* Business, finance, marketing and management
* Communications
* Computer hardware and associated technology
* Database design
* Man/machine interfaces and knowledge engineering
* Mathematics relevant to software, for example logic and set theory
* Operating systems and programming languages
* Software engineering and system design.

A.17 The goal of each new course development would be to obtain approval from the IEE and/or the BCS towards the accreditation of those who attend it. Accreditation should depend not only on approval of the syllabus and training

material, but also on conditions, such as the following:

- ⋆ There should be a reading list and a self-test procedure by which a professional can, without public embarrassment, check personal readiness to attend the course.
- ⋆ There should be method for assessment of individual students, with explicit pass and failure grades.
- ⋆ There should be advice and consultancy available for teams who find technical or other obstacles to putting into practice what they have learnt.
- ⋆ There should be a follow-up of students who have attended the course to check the quality of the instruction and its impact upon working practices.
- ⋆ The courses may be taught as evening classes, short residential courses, summer schools, or by distance learning techniques, whichever is most convenient for the students and their employers.

A range of professional examinations should be instituted, to check the efficacy of the education and record the achievement of those who pass them.

Targets

A.18 Targets for the in-service educational programme and performance monitoring should be maintained by STARTING, in collaboration with the IEE and BCS. We recommend also that the programme should assess the quality of education and its impact on working practices. The inititial target of the programme should be that a significant (for example 50%) and representative proportion of companies and organisations using IT and operating in the UK should be committed to the in-service educational programme by 1990. By then, participants should have increased their formal in-service education to:

Board and senior managers	1 week per year
Middle managers	2 weeks per year
Technical IT staff	4 weeks per year
Non-technical IT staff	2 weeks per year
Other professional staff	1 week per year

Funding

A.19 It is vital that in-service education does not constitute a hardship to those participating. Shortage of funding, qualified training staff and suitable courses is recognised. Consequently, considerable imagination must be applied to provide a cost-conscious programme of such wide scope and availability. This, in itself, presents an opportunity and justification to develop innovative training techniques. We recommend that the IEE and the BCS consider novel methods of disseminating

educational material (for example the Open University's distance learning and independent study techniques) and of examination (for example the Educational Testing Service of Princeton, New Jersey).

A.20 For companies, much of the cost of in-service education might be absorbed by some displacement of present in-service training; a national programme should allow economies of scale and lead to radically reduced costs. Government funding would be advantageous for rapid initiation of the in-service educational programme. Government should contemplate some direct or indirect training subsidy to support those setting up the new courses and the employers of those who attend them.

A.21 The in-service education programme is sufficiently important that those organisations making a significant contribution should receive public recognition.

13. Appendix B: Safety-critical software

The problem: non-technical

B.1 No computer software failure has killed or injured a large number of people. It is just conceivable that such a tragedy could occur. What steps should be taken to:

* prevent such a disaster,
* cope with it when it does occur,
* ensure such a disaster, having happened once, cannot recur?

The problem: technical

B.2 Stored-program digital computers must be among the most reliable mechanisms ever built by man. Millions of computers throughout the world are executing millions of instructions per second for millions of seconds without a single error in any of the millions of bits from which each computer is made. In spite of this, nobody trusts a computer; and this lack of faith is amply justified.

B.3 The fault lies not so much in the computer hardware as in the programs which control them, programs full of the errors, oversights, inadequacies and misunderstandings of the programmers who compose them. There are some large and widely used programs in which hundreds of new errors are discovered each month; and even when these errors are corrected, the error rate remains constant over several decades. Indeed, it is suspected that each correction introduces on average more than one new error. Other estimates offer the dubious comfort that only a negligible proportion of all the errors in these programs will ever be discovered.

B.4 New computers are beginning to be used in increasingly life-critical applications, where the correction of errors on discovery is not an acceptable option, for example industrial process control, nuclear reactors, weapon systems, station-keeping of ships close to oil rigs, aero engines and railway signalling. The engineers in charge of these projects are naturally worried about the correctness of the programs performing these tasks, and they have suggested a number of expedients

for tackling the problem. Many of these methods are of limited effectiveness because they are based on false analogies rather than on a true appreciation of the nature of computer programs and the activity of programming.

B.5 The steps which ACARD has been considering in answer to the introductory question, are discussed under the following headings:

* Disaster prevention
* Disaster management
* Disaster analysis

Disaster prevention

B.6 The initiative for disaster prevention must come from the UK government and system customers. Current software is built, operated and maintained using methods and tools which are not keeping pace with the development of the hardware, nor with the increased sophistication demanded by new applications; nor does it take account of progress of research into the reliability of programs. The necessary improvements in software engineering require investment in advanced development and production techniques, education, training and legislation. Legal obligations should be at least as stringent as those imposed by the Data Protection Act, and the care and time required for detailed drafting of legislation will be just as great. A start must be made immediately.

B.7 The remainder of this appendix outlines an imaginable solution that may emerge over the next fifteen years. It is intended to promote rather than to pre-empt a discussion of the details.

Registration

B.8 A register must be established of those (software) systems which, if they fail, will endanger lives or public safety.

Operation (demand side)

B.9 Before any organisation can operate a life-critical computer system it must first obtain a Licence To Operate (LTO), which will only be issued when the operator can demonstrate that certain conditions (detailed below) have been met.

B.10 Each life-critical system must be operated by a Certified Software Engineer who is named as being personally responsible for the system. This Certified

Software Engineer must have received the appropriate mathematical training in safety-critical software engineering.

B.11 A life-critical system must be adequately maintained; this must be one of the conditions of the LTO. Maintenance (that is, rectification and development) must be the responsibility of a named Certified Software Engineer.

Construction (supply side)

B.12 Only approved suppliers may be allowed to build life-critical computer systems; thus suppliers must gain a Licence To Construct (LTC). An LTC will only be issued to suppliers who can show that they build systems to certain approved standards using methods which are mathematically sound, and using safety-certified tools and staff.

Certification

B.13 An LTO must only be granted when a Safety Certificate has been issued. Certificates must be issued for limited periods, for example, five years. Operational systems will thus need to be recertificated (relicenced) periodically (analogous to Certificate of Airworthiness).

Reliability data collection

B.14 To aid research into system reliability, and to assist Boards of Enquiry, all registered life-critical software systems must supply operating data on the Licensing Authority.

Disaster management

B.15 In the past, the danger arising from failure of computer hardware and software has been limited by switching off the computer and reverting to manual operation if necessary. In future, there will be applications for which this fall-back procedure is not available. The computers will have to continue to run, and any necessary software changes and corrections will have to be inserted into the incorrectly running system. For these applications, specially stringent precautions are necessary.

Procedures

B.16 The Licensing Authority should require disaster management procedures to

be laid down in advance of operation and practised regularly during operation (that is 'fire drill practice'). The documentation of the system must meet a standard which would permit a team of experts/specialists to master it during the progress of an emergency.

Data logging

B.17 The disaster management procedures should include the logging of data so that any subsequent Enquiry can ascertain the progress and cause of the disaster (analogous to the 'black box recorder' in an aeroplane).

Emergency call-out

B.18 There must be more than one Certified Software Engineer available to the operating company; and a duty rota should ensure that one of them is always available at short notice. Procedures must be set up for calling out a team of expert specialists in a longer-lasting emergency.

Disaster analysis

Collection of reliability data

B.19 During the normal (safe) operation of any life-critical system, data on its performance and reliability must be made available to the Licensing Authority. This data will be made available to any Enquiry. (This is additional to the data logging required in para B.14.)

Board of Enquiry

B.20 Any disaster should be the subject of an official Board of Enquiry (similar to rail and air disaster enquiries). A Board of Enquiry must have the power to make changes to the system under investigation and/or the methods, tools, products and staff associated with the certification procedure.

Any error triggers Board of Enquiry

B.21 Any error, no matter how 'small', in a software system which has been certified as being safe must be the subject of an Enquiry. This is the only way of discovering weaknesses in the certification process itself, or misuse or misunderstanding of its application. Enquiries concerning non-fatal errors should not have

disciplinary implications, so that operators are encouraged always to give notification of minor faults.

Near miss

B.22 Any serious 'near miss' must be reported to the Licensing Authority. An Enquiry should be held if the Licensing Authority is concerned at the incident's implications.

Safety certification

B.23 The UK must develop the ability to certify safety aspects of software system construction and operation. These include:

- certification of the mathematical soundness of the methods of construction;
- certification that certified methods are properly applied during construction and subsequent maintenance (rectification and development);
- certification of the tools used during construction and maintenance;
- certification of the software engineers who build and maintain the systems;
- certification of the end product, that is, the software itself.

B.24 Methods should not be certified which are merely 'good practice'. Safety and reliability require more rigorous theoretical bases than existing good practice, so that system behaviour can be accurately and consistently *predicted;* hence the need for mathematical soundness to enable prediction to be based on mathematical proof.

B.25 Certification of a tool will only be given when it is shown that the tool preserves the mathematical soundness of the method it supports.

B.26 Certification of software engineers will only be given when they have completed an approved level of formal mathematical and methodological training together with an approved track record of experience. Certification should be of limited duration; recertification should require additional formal training both of the refresher type and new developments. Recertification should occur at regular intervals.

B.27 Certification of end products (and their components) implies proof obligations in addition to thorough testing. Proofs must be performed and checked by competent mathematicians or by a machine running certified software.

B.28 As in other branches of engineering, the rigour of the inspection procedures

should be adjusted to the degree of risk, the severity of the danger and the cost. For example, we can imagine the emergence of several levels of certification:

a. *Disaster Level.* Failure could involve more than ten deaths. The whole of the software must be checked by formal mathematical proof, which is itself checked by a competent mathematician. Further precautions required if damage limitation by switch-off is not feasible (para B.15).

b. *Safety Level.* Where failure could cause one death, but further danger can be averted by switch-off. The whole of the software must be constructed by proof-oriented methods, checked by a competent mathematician. On occurrence of a fatality, the mandatory Enquiry must name the programmer and mathematician responsible, who might be liable for criminal negligence. Perhaps one error per 100,000 lines of code would be a realistic expectation, so that most shorter programs will contain no errors.

c. *High Quality Level.* Appropriate for software sold commercially, where error could bring financial loss to the customer. By law, such losses should be reimbursed. All programmers involved should be certified competent in mathematical methods of software design and construction. Their use of the methods is checked by sampling. An acceptable error rate would be one error per 10,000 lines of code delivered. Each error corrected requires recertification at Safety Level. If the target error rate is exceeded, certification is withdrawn. Eventually, all software used to construct other certified software should be certified to this level; and the construction of 'disaster level' software should include independent checks on the correct working of support software used (for example, check of binary code against higher level source codes).

d. *Normal Quality.* Corresponds roughly to the best of current practice (say, one error per 1,000 lines of code). The methods used to construct software to higher levels of reliability may also be used to achieve normal reliability; and this should bring a significant improvement in programmer productivity and a reduction in the whole life cycle costs of the programs they produce.

14. Appendix C: Product standards and certification

The objective

C.1 This appendix discusses the opportunities and benefits of standardisation of the designs and interfaces of software products, for example programming language compilers, applications-oriented packages (graphics, statistics etc), operating systems, word processors, spreadsheets etc.

C.2 The objectives of standardisation should be solely to benefit the consumer. A good standard should provide:

- ⋆ stability and protection of investment in the face of rapid technological obsolescence;
- ⋆ opportunity to mix and match components, services and training, enabling purchase from suppliers who compete in providing the best value for money;
- ⋆ assurance that the selected product meets high standards of quality, reliability and fitness for purpose.

At present, the first point seems the most important, but in the future the second and third points should dominate.

C.3 Standardisation may also be used as a weapon in the commercial struggle between suppliers; and particularly between a dominant large supplier and its smaller competitors. This struggle does not necessarily contribute to the quality of the standard as perceived by the consumer.

The problem

C.4 In a rapidly developing technology, it is extremely difficult to establish the first standard for each new requirement as it emerges. There is a choice of unattractive alternatives:

- ⋆ *The standard is promulgated before its first implementation.* Such standards tend

to contain oversights, misjudgements and inconsistencies; they are unnecessarily expensive to implement and use. They are rarely appropriate to the technology that has emerged by the time the first implementations are available, and they cannot meet customer needs which come to light only after experience of use.

* *The standard is designed after several rival products have reached the market place.* In this case the standard has to be a compromise between the commercial needs of the rival manufacturers. Such standards will tend to be complicated, with many concealed inconsistencies; and they are unlikely to meet objective standards of quality as perceived by the customer.

C.5 *There are many standards which are first drafted before implementations exist and finally completed after several competing products exist.* Such standards can combine the disadvantages of both the above alternatives. It is not surprising that the first standards promulgated for software products are no more satisfactory than the first standards (say) for physical measurement of space and time in the Middle Ages. Yet the need for standards in software is so great that they are welcomed and promulgated with an enthusiasm totally unrelated to their intrinsic quality.

Towards a solution

C.6 In other branches of engineering it is taken for granted that standards must evolve in the light of experience and of improved understanding of the relevant scientific concepts and principles. The improved quality of physical and engineering standards owes much to the goal-directed research and advance development conducted in the great National Laboratories of Physics and Engineering.

C.7 Improvement in standards for software products will require a similar devotion of effort both to the development and application of the relevant mathematical theories and to experimental confirmation of the parameters of cost, effectiveness and customer satisfaction. Such an effort will require a central initiative and technical co-ordination on an international scale over a period stretching into the next century. It would be unrealistic to expect the effort to be wholly supported by the software supply industry, or by their customers.

C.8 This country has an outstanding record of achievement in the development of the mathematical concepts and theories relevant to the formulation of standards in Information Technology. However, there is at present an almost total gap between this theoretical work and the activities of standardisation committees, both at national and at international level. An equally dangerous gap is that which can arise between the standardisation committees and the experimental work necessary for design, validation, implementation, evaluation and certification of proposed

standards. To close these gaps will require a change in the expectations and policies of standardising authorities, and a very considerable increase in the amount of qualified technical effort expended on standardisation and on co-ordinated goal-directed research prior to standardisation. Attention must be paid not only to improvement of the quality of the standard, but also to the means by which customers and suppliers may migrate from existing to better standards. Furthermore, the body responsible for a standard must also devise and administer procedures for checking that products meet the standard, and perhaps issue a seal of approval like the BSI's 'kitemark' to products which meet measurable criteria of performance and reliability.

C.9 At present, suppliers are very busy in meeting the initial software standards which are now emerging, and customers are very busy in learning to live with them. Consequently, there is no immediate commercial pressure to improve the quality of standards for software products; there is even little realisation that such improvement may be possible. Indeed, the wide propagation of the first standard in each field is known to create considerable resistance to change, and even in the long term may prevent acceptance of improved standards. The layout of the typewriter keyboard is a good example of an improvable standard that has never been improved.

C.10 In spite of this risk, we recommend a policy of long-term, goal-directed research into software standards. This should be co-ordinated internationally, or at least on a European scale. If a single nation or group of co-operating nations can obtain a reputation for the quality of the standards they promulgate, this will give the local supply industry a head start in the international market-place. If a national certification agency gets a reputation for the value of its seal of approval, even further advantages will obtain. Finally, the research towards standardisation will develop skills and techniques in advanced international market research, as well as specification, design, prototyping and implementation of high quality software. If the standardisation research enterprise is properly managed, this technology will spin off to benefit local suppliers, not only of products but also of bespoke software, enabling them to secure a commercial edge in the international market place for all other products and services.

15. Glossary of acronyms

BASSMATT .....British Advisory Service for Software Marketing, Applications, Training and Technology Transfer

BCS.....British Computer Society

CAD.....Computer Aided Design

CAM .....Computer Aided Manufacture

CCTA.....Central Computer and Telecommunications Agency

DES .....Department of Education and Science

DHSS .....Department of Health and Social Security

DTI .....Department of Trade and Industry

IEE .....Institution of Electrical Engineers

IPSE.....Integrated Project Support Environment

ISF .....Information Systems Factory

IT.....Information Technology

MOD.....Ministry of Defence

NCC .....National Computing Centre

NEDO .....National Economic Development Office

R&D .....Research and Development

SPC.....Software Production Centre

SRCB .....Software Reusable Components Brokerage

STDC .....Software Tools Demonstration Centre

STARTING .....Software Technology and Applications Review Team of INdustry and Government

STARTS/PPG .....Software Tools for Application to large Real Time Systems/ Public Purchasers Group

TYPSSEA.....Ten Year Pipeline Strategy for Software Engineering and Applications

UK.....United Kingdom

USA .....United States of America

Printed for Her Majesty's Stationery Office by Linneys Colour Print, Mansfield
6/86 Dd 739462 C31 47701

The following ACARD reports are available from Her Majesty's Stationery Office –

The Applications of Semiconductor Technology (1978)	ISBN 0 11 630807 9
Industrial Innovations (1979)	ISBN 0 11 630808 7
Joining and Assembly: The Impact of Robots and Automation (1979)	ISBN 0 11 630810 9
Technological Change: Threats and Opportunities for the United Kingdom (1980)	ISBN 0 11 630812 5
Computer Aided Design and Manufacture (1980)	ISBN 0 11 630814 1
Biotechnology (1980) (with the Royal Society and the Advisory Board for the Research Councils)	ISBN 0 11 630816 8
Information Technology (1980)	ISBN 0 11 630818 4
Facing International Competition: The Impact on Product Design of Standards, Regulations, Certification and Approvals (1982)	ISBN 0 11 630783 8
The Food Industry and Technology (1982)	ISBN 0 11 630822 2
Improving Research Links between Higher Education and Industry (1983) (with the Advisory Board for the Research Councils)	ISBN 0 11 630784 6
New Opportunities in Manufacturing: The Management of Technology (1983)	ISBN 0 11 630823 0
Exploitable Areas of Science (1986)	ISBN 0 11 630827 3

Exploiting Invention (1981) is available from ACARD

HMSO publications are available from:

HMSO Publications Centre
(Mail and telephone orders only)
PO Box 276, London, SW8 5DT
Telephone orders 01-622 3316
General enquiries 01-211 5656
(queuing system in operation for both numbers)

HMSO Bookshops
49 High Holborn, London, WC1V 6HB 01-211 5656 (Counter service only)
258 Broad Street, Birmingham, B1 2HE 021-643 3757
Southey House, 33 Wine Street, Bristol, BS1 2BQ (0272) 24306/24307
9-21 Princess Street, Manchester, M60 8AS 061-834 7201
80 Chichester Street, Belfast, BT1 4JY (0232) 234488
13a Castle Street, Edinburgh, EH2 3AR 031-225 6333

HMSO's Accredited Agents
(see Yellow Pages)

and through good booksellers

World demand for Information Technology (IT) products is large and growing rapidly. The market for IT-related software alone is worth some 40 billion dollars a year, and annual growth is about 30%.

The possibility that the UK might be failing to develop and apply IT-related software sufficiently quickly and widely - and that Britain could therefore be losing its expected share of the world market - is examined in this ACARD report.

The Advisory Council's detailed analysis of the challenge to the UK software industry is accompanied by strategic targets and tactical recommendations. All are designed to offer companies, government departments and educational establishments the clear picture they need if they are to develop a coordinated approach to developing, marketing and applying effective software products.

£6.00 net

ISBN 0II 630829 X