RESTRICTED

SCIENCE AND ENGINEERING RESEARCH COUNCIL
RUTHERFORD APPLETON LABORATORY

INFORMATICS DIVISION

SOFTWARE ENGINEERING GROUP NOTE 113

BASSMATT
Further Thoughts
(draft 1)

issued by
R W Witty
26 March 1986

DISTRIBUTION: D E Talbot
R Witty, RAL
EL/DTI/BASSMATT

1. INTRODUCTION

Here are some more thoughts about BASSMATT, including extracts from the ACARD report and the Alvey SE Strategy to put BASSMATT into context.

CONTENTS

2. EXECUTIVE SUMMARY OF ACARD REPORT

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. UK 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 coordinated 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 cooperation 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.

3. THE ROLE OF GOVERNMENT

3.1 THE PUBLIC PIPELINE

The UK Government plays an influential role in all aspects of IT. Government 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.

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 1).

```
Research        )
Development     )  Public R&D
                        |
                        |        Development-Procurement Gap
                        v
Requirements    )
Purchase        )  Public Procurement
                        |
                        v
Use             )
Maintain        )  Public Use
Decommission    )
```

<u>Figure 1 The Public Pipeline</u>

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 2-3 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.

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 then cannot be recovered because it is either too expensive or impossible to win back these markets.

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.

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.

3.2 THE DEVELOPMENT - PROCUREMENT GAP

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 focussed 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 marketplace.

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.

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.

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.

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.

3.3 MAIN GOVERNMENT RECOMMENDATIONS FROM ACARD REPORT

TYPSSEA : Improving the Public Sector Software Pipeline

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 coordinated by an inter-departmental, long term strategy. This strategy is provisionally named TYPSSEA - the Ten Year Pipeline Strategy for Software Engineering and Applications.

TYPSSEA should have four main elements:

* Public purchasing to exercise demand side leadership
* Technology transfer acceleration
* Public sector R&D continuity
* Public sector in-service training

Public Purchasing and Procurement

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.

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

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 should be accelerated.

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

Public Sector Research and Development

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.

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

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.

We recommend that, through the TYPSSEA plan, coordination 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

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.

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

4. RATIONALE FOR TECHNOLOGY TRANSFER

4.1 Customer Factors Limiting Innovation

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."

"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]

"A number of policy perspectives may be deduced:

* <u>There is a need to stimulate a high rate of experimentation in the IT field</u>

  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.

* <u>Knowledge acquired by the consumer is the key to rapid acceptance</u>

  Much can be done by Government to create awareness of IT services and technological opportunities. The analysis contained in this report[4] 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.

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]

4.2 User-Applier-Supplier Symbiosis

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.

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.

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.

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

5. FOREIGN INITIATIVES

5.1 The United States of America

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.

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.

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, was formed by twelve aerospace and defence companies to achieve technology transfer.

* The DOD Software Engineering Institute (SEI) is based at Carnegie Mellon University and will last at least 15 years with an initial $100 million over 5 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.

5.2 Japan

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.

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:

* 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.

5.3 France

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.

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.

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.

6. BASSMATT: TECHNOLOGY TRANSFER ACCELERATION (FROM ACARD REPORT)

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 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.

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. 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.

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.

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.

BASSMATT's functions and effort should be as follows:

* (30%) Technology Transfer
* (25%) Independent technical advice
* (15%) Marketing Support
* (15%) Collaborative Research
* (10%) 'After Alvey' Programme support and collocation
* (5%) STARTING and TYPSSEA support

6.1 Technology Transfer (30%)

BASSMATT should speed the flow of technology transfer between the use/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 Reusable Components Brokerage (see below) as suggested in the Alvey Report[7].

* 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 Reusable 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 reusable 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
    - host short courses
    - investigate the quality of educational courses
    - investigate the efficacy of professional certification

6.2 <u>Independent Technical Advice (25%)</u>

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 of new software engineering methods, tools and products including helping STARTS/PPG tools initiative. Cost benefit studies.

* 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.

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.

6.3 Marketing and Legal (15%)

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.

6.4 Home to the 'after Alvey' Research Programme (10%)

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.

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

6.5 Research (15%)

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.

6.6 STARTING Support (2%)

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 of STARTING of evidence from industry, academia and government.

6.7 TYPSSEA Support (3%)

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 interdepartmental contact to help formulate objectives, strategy and tactics.

6.8 Organisation

BASSMATT should:

* have an initial 15 year lifetime with 5 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 5 yearly review should recommend its closure.

7. DTI RECOMMENDATIONS FROM ACARD REPORT

7.1 TYPSSEA

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

7.2 Demand Side Leadership

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.

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

7.3Technology Transfer

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 9.19).

* Increased support for the STARTS/PPG software tools initiative.

* Continue support for 'awareness' programmes

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

7.4 Support for Product Innovation, Marketing and Exporting

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.

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.

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.

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

7.5 Exporting

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, eg 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, eg design rule checker should be able to switch between BSI/DIN/ANSI/ISO versions of requirements if overseas market requirements are country specific, eg 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.

7.6 Further Study

<u>We recommend</u> 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.

* the legal and technical issues surrounding safety-critical software, ie software products whose (faulty) operation could endanger public safety.

* the potential of software product quality certification (the 'software kite-mark')

* the role of new technical standards as a means of helping the UK industry tackle the world market

* the role for the certification of practising software engineers

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

7.7 Fiscal and Legal Recommendations

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

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; about advising exporters on legal problems associated with selling software abroad.

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.

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.

8. BASSMATT - SECOND LEVEL OF DETAIL

Some of the issues to be discussed include

1. Exact terms of reference (objectives, charter, mission etc)
2. Budget
3. Management (external and internal)
4. Location
5. Staffing
6. Policy
7. Technical Programme
8. Equipment
9. Relationship to other UK government activities
10. Relationship to UK industry
11. Relationship to EEC
12. Relationship to foreign countries, firms, etc
13. Legal Status
14. Legal Issues
15. Name - need something better than BASSMATT?
16. Subcontracting
17. Criteria for success and failure.

8.1 <u>Terms of Reference</u>

To be based on ACARD recommendations. Need to resolve whether BASSMATT should include the Research centre of excellence, which has been mooted by MOD, or not. Collocation or incorporation of the research centre with BASSMATT would be sensible if building a national institute.

8.2 <u>Budget</u>

The questions are

a. How much?

b. For how long?

c. From whose pocket(s)?

Based on the assumptions of

a. 20 year lifetime

b. 150 professional staff + 150 support staff

c. world class equipment and facilities

d. assuming use existing building (perhaps nodified and updated) to reduce capital costs.

FYFL

	1	2	3	4	5
Staff Prof Support	 1.2 0.6	 3 3	 6 6	 9 9	 9 9
Equipment Capital Recurrent	 1 1	 5 1	 5 1	 3 2	 3 2
Recurrent	1	5	10	15	15
Capital (non equipment)	1	5	2	1	1
TOTAL £M	5.8	22	30	39	39

RECRUITMENT PROFILE

	1	2	3	4	5
Professional numbers cost @ £60K	20 £1.2M	50 £3M	100 £6M	150 £9M	150 £9M
Support numbers cost @ £60K	10 0.6M	50 £3M	100 £6M	150 £9M	150 £9M
TOTAL numbers cost	30 £1.8M	100 £6M	200 £12M	300 £18M	300 £18M

8.3 Management

The questions are

a. to whom will BASSMATT report (external)?

b. how will BASSMATT be organised internally?

My suggestions are that BASSMATT should be organised like a company with a Board of Directors representing the shareholders (ie those government departments putting in the money) and some interested parties (eg industry, academia). One or two 'independents' might be desirable. (Industry means manufacturing and service industries not just G Holmes et al!)

BASSMATT should have a Chairman; a full time appointment who runs the Board meetings and is the main political link to the outside world. He should be assisted by a Managing Director who runs BASSMATT internally. The rest of the internal structure will come naturally from the functions.

```
Government, Industry, Academia (ie the 'community)
    |
    Board
     |
     |-- Chairman   )
     |              ) BASSMATT full time
     |-- MD         )
     |
     |-- Manufacturing, Service, IT industry reps
     |
     |-- Government Dept reps
     |
     |-- Academia
     |
     |-- Independents
```

Who hires/fires the Chairman and MD?

8.4 Location

It is plainly stated in the ACARD report that BASSMATT should be located in the M4 corridor between London and Bristol. BASSMATT needs to be close to the bulk of the community it serves. One day politicians will learn that Silicon Valley did not happen as a result of regional development policies!

The choice of location raises some issues.

1. It must be attractive both in terms of general geography and working conditions if we are to attract the right staff.

2. Good working conditions means good buildings, decoration, landscaping etc.

3. Will a new building be constructed (£20M?) or will existing accommodation be refurbished? If the latter then the government candidates are

   DTI/NPL
   SERC/Rutherford
   AEA/Harwell
   AEA/Aldermarston
   Any others?

Clearly premises could be rented from the private sector but they will not have the general infrastructure and environment that is needed to make up a stimulating laboratory environment. An industrial company could act as host but no doubt technology would flow, or be felt to flow, preferentially into the host company no matter how high any 'Chinese wall'.

8.5 Staffing

Based on experience at Atlas Lab, Rutherford Lab and DOD/SEI it is likely that BASSMATT will need 150 IT professionals supported by 150 other staff such as secretaries, administrators, computer operators, junior computer programmers, information services operatives (database maintenance, query services etc) etc.

Unless Government is committed to hiring really good people then BASSMATT should not be born. The NCC has shown that the wrong location and poor salaries are a recipe for a second rate institute. BASSMATT will only succeed if it can win the technical respect of a wide community; good staff are the key.

Staff can be obtained by

a. Permanent posts.

b. Fixed Term contracts.

c. Secondment from industry, academia, government departments.

d. Host location for sabbaticals.

I would favour a mixture of all four. One needs permanent staff to provide continuity of policy and operation, give stability, take long term view which is important as many activities will have 5-10 year payback periods.

However not all staff should be permanent. Fixed term contracts of say 3-5 years, perhaps renewable on 2 year rolling basis, will allow a flow of good people which is necessary too. The flow in will bring in new ideas and experience and the flow out will transfer technology 'on the hoof'. With the high rate of turnover of IT Staff fixed term appointments are feasible, especially for brighter, younger staff who will likely use BASSMATT as a training ground route to higher salaries - we should consider this part of BASSMATT's training function! Fixed term contracts could also pay companies to release people to work at BASSMATT (subsidised secondment). This should be a way of getting experienced people from idnustry and academia.

8.6 Policy

Lots of policy decisions need sorting out up front so as to influence the Terms of Reference eg

1. Is BASSMATT a secure or open establishment?

2. Is any military secret work done?

3. What level of commercial confidentiality will be handled?

4. What is the policy on openness towards EEC, NATO, European companies, USA companies, Japan, rest of world? Will BASSMATT take foreign secondees?

5. What about Intellectual Property issues. The DOD/SEI is right to make this a separate activity.

8.7 Technical Programme

The easy bit!

As laid down in the ACARD report. Clearly needs refining if the funding departments have special requirements. Will evolve over time.

8.8 Equipment

The level of equipment has to be world class. This is necessary to do the job at the right level. If the equipment investment is 'serious' then this will be a major attraction to the best people BASSMATT needs.

The equipment needs will be a function of time as new machines are always coming onto the market. A long term policy will be needed to insulate BASSMATT from this trend (cf SERC's Common Base Policy).

Today I would plan for a single user system for each professional plus two per three support staff. BASSMATT will also need large'mainframe' type capacity and good network connections just to do its everyday job. It will also need something separate that visitors can play with (ie the SPC 'showcase').

150 professional SUS	150 x £25K =	£3750K
2/3 x 150 support SUS	100 x £25K =	£2500K
Big VAX for services and support		£ 500K
Big VAX for development work		£ 500K
VAX + SUS for 'Showcase' use		£1500K
Networking		
Local and Wide Area		£ 500K
Other (including documentation typesetters, facimile, video et		£ 750K

		£10,000K

The equipment will need constant updating and enhancement.

8.9 Relationship to Other UK Government activities

ACARD recommended collocation of 'After Alvey' and BASSMATT. There needs to be a policy decision as to whether BASSMATT is seen as remaining as a technology transfer outfit, or whether it should be the nucleus for a major new national centre for IT (as DOD/SEI intends).

Not until we have a view on this question is it worth getting down to details.

Obviously existing things such as the present Alvey Programme, STARTS, NCC/STDC etc can be fitted in with BASSMATT, especially the NCC/STDC which is mentioned explicitly in the ACARD report.

8.10 Relationship to UK Industry

Issues are

1. What does industry get from BASSMATT?

2. Does industry know it needs BASSMATT?

3. Will a seat on the Board be enough?

4. What does industry see as the problems/advantages of BASSMATT eg IPR, leakages of information between companies.

5. Is BASSMATT a threat to consultancies and body shop outfits? No! It does things which are either medium term, not being done by industry or helps such people win business.

8.11 Relationship to EEC

1. Do we have to get EEC permission to set up BASSMATT (like we did for Alvey)?

2. BASSMATT, especially if collocated with After Alvey, makes a natural centre for coordinating UK technical contributions to European initiatives, but see policy decision on breadth of role.

3. Does Europe need such a centre of technology transfer? Could BASSMATT attract European funds? Perhaps this defeats its purpose which is essentially national although we need to build a European industry if we are to survive the USA and Japanese onslaught.

8.12 Relationship to Foreign countries, firms etc

Policy decisions will be needed about how we handle relationships with foreign countries. Clearly the DOD/SEI was not going to let anything out into the outside world if they could avoid it - hence their 'secure' mode of operation and legal project. Conversely they were deliberately setting out to suck in as much foreign technology as possible; they have already enticed one Alvey software engineering project leader to spend a summer in Pittsburgh to do a brain dump.

The Japanese information diode is well known, too. I guess the policy must be similar to DOD/SEI - we must fight the economic war according to the USA and Japanese rules, not those of UK academics.

Foregin companies raise another, standard problem ie how do we handle IBM(UK), H-P(UK), DEC(UK) etc? I am not aware of any government policy to help us with this question so presumably we must fill the vacuum. Given that IBM (with Z) and H-P (with OBJ) for instance are sucking in technology from UK academics the issue is much wider than just BASSMATT and I feel it is high time that DES sorted out the academic IPR issues properly so that academics are not free to sell their (UK funded) research to whichever foreign company or government they feel like. Academic freedom can act against the national interest when UK companies do not support UK universities as well as foreign companies in both gifts, sponsorship and technology take up.

8.13 Legal Status

If BASSMATT is just another branch if a government department then it will fail. Poor salaries and bureaucracy will mean it will never get off the ground.

What form can BASSMATT legally take? It cannot 'trade' because it will be 100% government funded, so 'privatisation' is a non-starter. Looks like some kind of 'quango'. Can these be tolerated today?

8.14 Legal Issues

I suggest that BASSMATT has a 'legal project' just like DOD/SEI. It could be subcontracted. BASSMATT needs to get IPR issues etc well understood. The use of cocom, embargoes, extra-territorial legislation etc by foreign countries to hamper our competitiveneness needs to be tackled explicitly by lawyers not software engineers.

8.15 Name

BASSMATT was always meant to be a provisional name. Perhaps a better one can be found. Such things are actually quite important when one is trying to build something from scratch which needs to attract good staff.

Perhaps the first Chairman and MD could have the privilege of chosing the name and getting it approved by the first Board meeting?

8.16 Subcontracting

Some functions should be subcontracted. The legal questions are a good example because they are distinct from the actual tecnology and therefore do not need the BASSMATT environment. Serving geographical outposts, in which mini-BASSMATTs serve outcrops of industry are another candidate. NCC/STDC can serve 'the North' from this point of view.

Subcontracting should be used sparingly and deliberately. If it is misused it will defeat the whole point of one physical concentration of skill and activity.

8.17 Criteria for Success and Failure

The corollary to the Terms of Reference should be an explicit statement to enable success and failure to be judged. I do not know what they should be - obviously the sentiment of the community will be the ultimate test but something more concrete should be attempted.

Perhaps a 'Doomsday survey' of the use of software and IT in manufacturing and service industries (as STARTING might request) should be undertaken at regular intervals so that BASSMATT can attempt to judge if the rate of technology transfer has increased.

STARTING (see ACARD report) wants to set targets and suggests that BASSMATT should play a role in measuring the achievements against these targets so the targets should include something about BASSMATT thereby getting one mechanism do two jobs.

Targets could be set for:

1. Number of recipients of mailshots, tools guides.

2. Number of visitors to 'showcase' SPC.

3. Number of methods and tools 'sold'.

4. Number of components held and accessed in SRCB

and so on.

BASSMATT must not stagnate or outlive its usefulness. One sees these states occur in all reasearch labs and establishments when the original mission has been fulfilled and the people moved on or got stagnant. It must be clearly stated in the terms of reference that BASSMATT can fail because if it does not do its job well enough or 'fail' because its job is done (ie close down necessary) or fail because it has not moved with the times.

NOTE: Further relevant details are given in Appendix D, which include the Software Production Centre, Software Reusable Components Brokerage, the IPSE/ISF and examples of the services and equipment needed for these activities.

9. ROLE OF UK INDUSTRY

What sjould be industry's role in BASSMATT? Should it contribute funds directly or by levy (NO!) Should it contribute by giving its people free ie by secondment to be trained etc (YES). Should industry be charged for BASSMATT's services (NO - this will just increase paperwork and reduce customer base thereby defeating objective). Should it help set policy and targets (YES - seat on board but probably management should have consultative/advisory user groups; not a lot of committees please!)

10. TECHNOLOGICAL DEVELOPMENT

10.1 THE ENGINEERING APPROACH

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.

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.

10.2 QUALITY

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.

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.

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.

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 the 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.

10.3 PRODUCTIVITY

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.

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.

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

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.

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.

Integrated project support environments (IPSEs)[7] 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 address this deficiency and that IPSEs should form part of the UK's software research programmes.

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.

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.

10.4 SKILLS, EDUCATION AND MOBILITY

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"[5].

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.

The UK has a well known shortage of IT skills at all levels. 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 (ie, 1990) is to educate and train members of the current IT population.

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.

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.

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.

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.

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 .

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.

10.5 STANDARDS

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

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.

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.

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.

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.

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.

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).

11. REFERENCES

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

2. Information Technology Economic Development Council, NEDO.

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

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

5. "Submission to ACARD Working Group"
Institution of Electrical Engineers, 1985.

6. "Signposts For The Future"
IT Skills Shortages Committee
Final Report
DTI, 1985.

7. "Alvey Software Engineering Strategy"
Alvey Directorate, DTI,
November 1983.

12. APPENDIX A

EDUCATION AND PROFESSIONAL QUALIFICATION

THE PROBLEM

A.1 The key to UK success in Information Technology 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 oriented 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 formal education in software engineering more than once in his 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 (eg the IT Skills Shortages Committee) 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 cooperation 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 (ie 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.
* Data base design.
* Man-machine interfaces and knowledge engineering.
* Mathematics relevant to software; eg 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 course, 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 initial target of the programme should be that a significant (eg 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 (eg the Open University's distance learning and independent study techniques) and of examination (eg 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 amongst 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 by 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, weapons 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 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 do they 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 preempt 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 (ie 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). A 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 A LTO must only be granted when a Safety Certificate has been issued. Certificates must be issued for limited periods, eg 5 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 to 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 practiced regularly during operation (ie '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 Callout

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, 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 & 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, ie 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 ten thousand 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 (eg check of binary code against higher level source codes).

(d) Normal Quality

Corresponds roughly to the best of current practice, (say one error per thousand 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 coordination 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 coordinated 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 coordinated internationally, or at least on a European scale. If a single nation or group of cooperating 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. APPENDIX D

ALVEY SE STRATEGY (MODIFIED EXTRACTS)

1. GOALS AND OBJECTIVES

1.1 The Central Goals

Future IT products can be expected to be more complex than those of today and thus to place greater demands upon the people building them. The IT industry must meet this challenge, even though there is a growing recognition that system development techniques are inadequate for the large systems of today, let alone those of tomorrow. Skilled programmers are a scarce resource which is not being used efficiently. The IT industry is fragmented by organisation, by language and by target computer. One result of the consequent lack of commonality of environment or concentration of resources is that many programmers are not provided with even the simplest programming aids, let alone sophisticated ones. The economies of scale necessary to justify their introduction have not been perceived to exist. Software engineering may be considered as having two major goals for the future:

* improved quality ie satisfying criteria such as performance, reliability, security, on-schedule delivery and meeting the needs of the user;

* improved productivity ie reducing cost, not just of the development but of the life-cycle as a whole, including maintenance and future evolution.

Current software practice is centred on the programming process, and depends strongly on the skills, experience and resources of individual workers. Significant problems frequently result from inadequate effort being devoted to the front end of a development, notably concept formation, requirements definition, and design. Although there have been some efforts to study these problems, as well as interesting advances in both design verification and code verification, relatively little work has been devoted to integrating all of the stages into a common framework useful in production environments. Significant improvements in software productivity will be achieved when the current practice of repreated 'reinvention of the wheel' is replaced by the widespread re-use of prefabricated components. In the future then, software practice will tend to focus more on project management methodology, design, and component reuse and less on individual programming skills.

System design must include not just software design, but also hardware considerations. A narrow view of software engineering as just a collection of techniques to produce efficient software is not adequate. Software engineering should be aimed at the development of high quality systems, ie reliable, secure, efficient and easy to use, in a way that integrates hardware and software-based design criteria. In the future it must become information system engineering, not just software engineering.

1.2 Major Objective

To help achieve the general goals of improved Quality and Productivity the Software Engineering component of the Alvey Programme is focussed towards a strategic goal - that in 1989 the UK should be a world leader in Information System Factories (ISF). This goal is highly ambitious and competitive, as are the goals of the Japanese 5th Generation Project, the Japanese SIGMA project, the USA's SEI, STARS, MCC SPC initiative. The ISF objective implies a series of sub goals both in technology and timescale. The Alvey Software Engineering component will be judged on its ability to show that UK industry has increased both its software development product quality and software productivity as a result of striving to achieve the ISF. The strategy given in this document outlines the route towards the ISF with planned interim spin-offs so that quality and productivity gains may be achieved prior to the emergence of the ISF.

What is meant by an Information System Factory? Today, the production of most application-specific hardware/software systems - such as a banking network, a corporate management information system or production control system - does not in general make great use of development tools. In that sense it is not capital intensive. The application-specific part of the Information Technology industry is characterised as a cottage industry. It is predicted that it will not remain so for long, indeed the Japanese are already building 'software factories'. That to stay competitive in producing large, reliable, application-specific systems, IT companies will have to make a large investment in some kind of production facility. Exactly the same criteria will apply to manufacturing software products. This expensive facility - part hardware, part software, part stored knowledge - is an Information System Factory.

1.4 What Will Happen in Any Case

The Alvey SE strategy is based on the prediction that the production of application-specific information systems will cease to be a cottage industry and become a capital-intensive industry by 1990. The main reason has to do with software quality, in the widest sense of the word. Expectations of software quality, both within the industry and without, are very low. Today, programmers expect to have lots of bugs in their code, and the public expect computers to send them stupid invoices. This situation is not confined to the UK; it is worldwide. British standards of software quality are relatively high, while low in an absolute sense. This situation cannot last indefinitely. In the hardware field, one manufacturer (Tandem) has grown spectacularly by offering high reliability at a premium. This has been done against a background of hardware from IBM and others which is already highly reliable. The incentives to do the same in software, and the potential payoffs, must be much higher given the current poor quality of software. It seems highly likely that someone soon will "do a Tandem" in software, and either keep the method to himself or sell it very expensively. The Japanese are certainly trying, as are the Americans and the French. Without concerted action, the UK is bound to become an importer of this technology. If the UK is prevented from importing such technology then the industrial consequences could be very serious.

A number of other current trends are leading towards the 'capitalisation' of the software industry - the growing complexity of software systems, which demands new techniques and computer assistance to manage it, the dawning awareness of the importance of project and programming support environments, and the emergence of software packages which demand new skills to integrate them in particular applications. Finally, there is the emergence of non-Von Neumann architectures and VLSI, which are inevitably mixing the software and hardware design problems, making both more complex. All these are creating larger and more complex problems, which cannot be solved without a radically new level of automation and mechanical assistance.

2. THE CHANGING NATURE OF SYSTEM DEVELOPMENT

2.1 Summary

The expected changes that will result in the most significant increases in cost-effectiveness of software development over the next ten years are the following, listed in approximate order of expected impact. In the short term

1. incremental changes in programmer productivity through the more widespread use of design methodologies and tools

2. the coming together of methodologies and tools for the entire development life-cycle within integrated project support environments (IPSEs)

3. growing standardisation of development methodologies as a consequence of 2.

4. further refinement of suitable high-level programming languages appropriate to the integrated development methodologies.

5. growing interest in, and use of, formal specification methods and extension to animation.

6. automatic software generation techniques in limited form, probably first in the area of commercial systems built around Data Dictionaries. In the medium term

7. spread of powerful networked, personal workstations.

8. consolidation of the use of formal specification methods coupled with verification and growth in use of (semi-) automatic software generation.

9. development of reusable software and hardware modules, rigorously tested and formally documented.

10. second generation IPSEs adapted to support activities 8 and 9 above, coupled with greater use of higher-level languages.

And in the longer term:

11. the consolidation of the developments above into Information System Factories, coupled with the use of Intelligent Knowledge Based Systems, to provide 'automatic' assisted system development from user requirements expressed in high-level terms appropriate to the application rather than the implementation.

The crucial, and inter-related, technical developments underlying those changes will be:

1. integrated system (software and hardware) development methodologies supported by programming tools, administrative procedures and management information in an integrated environment.

2. formal specification, leading to 'animation' and verification.

3. reusable software and hardware components

4. automatic software generation

5. measurement and quality assurance and certification

These are discussed more fully below.

2.2 <u>Integrated System Development Process</u>

One view of the system development life-cycle is the following:

REQUIREMENT SPECIFICATION		
OVERALL DESIGN	typically costs	
DETAILED DESIGN	20-50%	
CONSTRUCTION	total development	
TESTING	budget	
OPERATION	- not usually quoted	
RECTIFICATION	euphemistically	
&	called	typically costs
EVOLUTIONARY	maintenance	50-80%
DEVELOPMENT		

Many design methodologies and software tools exist and are in sporadic use today, but the state of the art leaves much to be desired. For however good some of the tools may be, there are two serious problems. First, they do not support a development methodology or capture any data relevant to the management of the development process. Second, most tools support coding activities but fail to support the life-cycle in its entirety, or even fail to be compatible with other relevant tools. There is a need for more tools to assist with software specification, design, testing, rectification and development, as well as with management of software projects; and there is a need to integrate them into a coherent life-cycle support environment built on a database.

Recently there has been widespread recognition of these problems, with a resulting effort to develop better tools; a prime example is the growing work on the Ada Programming Support Environment (APSE), which should lead to a qualitative and quantitative improvement over today's state of the art. Viewed in the wider context of software engineering advances generally, two important short term benefits from such work should be increased programmer productivity in the technical tasks of project development and increased management awareness and control, leading to better decision making and costing. Moreover, the growth in use of integrated project support environments (IPSE) should provide the framework within which subsequent advances, such as improved specification and verification methods, can take place. This last point argues for a need for flexibility in IPSEs. They must not be closed systems incapable of accommodating improved techniques as these are developed elsewhere.

Whilst there are a number of issues still under debate, there does seem to be fairly widespread agreement on certain key characteristics that these environments will display.

First, and of crucial importance, there will be far less emphasis on the actual source text of the program than there is at present. Typical current practice focuses far too much attention on the source code representation of a program and far too little on other representations - expressions of requirements and various levels of specification. The tools which are most commonly employed are those concerned with manipulating and testing the source code representation. Yet most software projects that are 'unsuccessful' by some measure have already gone irretrievably wrong by the time that the first line of source code has been written. If there is to be real progress on the issues of effectiveness and cost then attention must be shifted from the code to requirements and design, and projects must be far more concerned with the 'higher level' representations. (Note that such a shift of attention is entirely compatible with an approach which emphasises re-use of existing components rather than always developing everything from scratch.)

Second, the environments will support a high degree of project visibility and traceability. At any stage of a project all relevant information will be readily available and there will be a proper basis for measurement of progress and detection of problems. For any identifiable activity there will be a record, not only of the end product of that activity, but also of the decisions (both positive and negative) which were taken during that activity.

Third, the environment will support various kinds of control. Management control, access control and configuration control all play an important part in addressing software effectiveness and software costs.

Reviewing the three issues above - emphasis on 'higher level' representations, visibility and control - leads to an inevitable conclusion: any given project employing such an environment must follow a defined methodology. This is not to say that the environment offers only a single methodology, but it is necessary for any given project to employ some defined methodology, and it is necessary for the supporting environment to 'recognise' this methodology (or at least certain aspects of it). Really, it is the methodology which addresses the issues of software quality and cost. The degree to which these issues are addressed depends upon the quality of the methodology and how well it is supported.

2.3 Formal Specification

The first qualitative change that will occur in system development will be the use of formal specification techniques. It is a large leap from today's practice to automatic program generation on a large scale, to proving theoretically that systems meet their requirments, to easy re-use of system components; but in each case the first step is formal specification.

Today's functional specifications are written in English, often with a liberal sprinkling of design detail in the difficult parts. Specifications written in natural language have the major defects that:

a. they are imprecise, ie are subject to conflicting interpretations
b. they may be logically inconsistent without the fact being apparent
c. they are apt to be incomplete

d. they cannot be used for (mechanically assisted) formal reasoning.

Use of natural language does not force the specifier to be precise at all times. In some cases he may be unaware of imprecision, which thus slips through; in other, he may decide to gain precision and by default the method chosen will probably be to take some design decisions and specify the requirement in terms of an implementation. Neither result is satisfactory.

The development of formal specification techniques should ultimately overcome these difficulties and lead to complete, precise specifications which do not contain any unnecessary design detail. Experience has already shown that efforts to translate natural langauge specifications into a logical form show up inconsistencies, ambiguities and omissions.

During the development of the complete specification, particular specifications can be 'animated' in the sense that their logical consequences can be explored. Questions such as What will happen if..? can be answered precisely, and the specification improved or modified as appropriate. In this way, purchaser and supplier can gain the clearest understanding of the system requirements. Another use would be simulation of critical aspects of the system, for example the user interface of a Command and Control System, so as to give the customer an early understanding of them.

Ultimately, formal methods can provide a very clear contractual basis for the statement of requirements and thus help to avoid disputes about whether the system meets the requirements or not. Numbers are hard to come by, but it is probably fair to say that most computer systems have to change after only a limited period of operation because the true operational requirements are, with the benefit of hindsight, perceived to be different from those originally requested. It has been argued that that is not so much a problem as an inherent characteristic of the real world which must be catered for in the development process. Systems must be designed to be capable of evolution. A rigorous path from specification through to implementation, with all the steps recorded, is essential if newly understood requirements can be fed in again at the beginning of the process without requiring a complete rewrite of the system.

2.4 Re-usable System Components

Today, hardware is thought of in terms of components whose behaviour is well understood and which can be put together in a number of ways to build a system. In the future, software will more and more come in packages until it too can be regarded as providing a set of component parts out of which software or mixed hardware/software systems can be constructed.

A number of trends will work together to bring this about. First, packaged software will account for a higher proportion of software sales to meet the enormous need for inexpensive software for personal, home or other small computer systems. Tailor-made software will be too expensive for this market and suppliers competing to reduce production costs will find it necessary to use mass production techniques, eg standardised design techniques, specialised tools, integrated project development environments.

Second, skilled programmers are a scarce resource and will continue to be so. Development techniques which provide a path away from today's labour-intensive methods will permit levels of production control and documentation adequate to the development of truly re-usable software. Third, design by components appears to offer the only solution to the problem already encountered today that some systems are so large and complex that their operation is hard to comprehend, their performance impossible to predict and their design impossible to optimise. Design in terms of components may permit only theoretical sub-optimisation but in practice this may be vastly superior to what could be obtained otherwise; and the ability to predict performance and cost, in advance of implementation, will be a major benefit.

2.5 Automatic Software Generation

Automatic software generation is in use now in a limited way, and is a very powerful technique for producing commercial software of the type that consists of simple, repetitive processing applied to a complex database. The development of data dictionaries in the commercial sphere, the trend to put the structure of applications into the database rather than into the programs, will encourage automatic software generation so that it can be expected to be a common technique in transaction processing within five years.

The experience thus gained, coupled with advances in specification techniques and the availability of a wide variety of software components, will subsequently enable automatic software generation to be applied to increasingly more complex processing tasks. It is here that Intelligent Knowledge Based Systems can be expected to make their greatest contribution to Software Engineering, in particular in determining and enforcing consistency of specifications and of the transitions from requirements specification to design and from design to implementation.

2.6 Measurement and Quality Assurance & Certification

Today it is difficult to predict the costs and timescale of a software development project, to measure the progress and productivity of the project team and to measure the quality of the finished product. The widespread adoption of integrated project support environments built on database technology will facilitate new research on the quantitative aspects of software development. A significant improvement in management effectiveness, productivity and product quality will occur when respectable metrication is introduced into the software development process.

Quantitative assessment of the benefits of new tools and techniques will provide a major stimulus to the introduction of further new techniques and further research, as hard-headed senior managers will be more easily persuaded to make the necessary investment funds available when presented with reputable, quantitatively argued cases with measurable pay offs.

Metrics and formal methods are the keys to effective quality assurance. The developers of good quality assurance techniques will enjoy a significant commercial advantage. The current shaky reputation of software will mean that the ever broadening range of customers will gravitate towards products bearing something akin to the BSI kite mark for simple products and components. Customers wanting more sophisticated products will favour suppliers who can offer independent, top class quality assurance and certification as part of the legal contract.

3.3 Integration

The second major need identified is for Integrated Project Support Environments (IPSE). The common understanding of an IPSE is that it should contain a compatible set of specification, design, programming, building and testing tools, supporting a development methodology that covers the entire life-cycle, together with management control tools and procedures, all using a central project database. That is already a very demanding requirment, exceeding that of the Ada APSE, but even then it does not go far enough. It does not cover multiple-language development; it does not cover mixed hardware and software development; it does not cover reusable components. There is certainly no agreement that one particular programming language meets all foreseeable needs, though there are individual proponents of this view for different languages. There is also considerable investment in software in the languages of the 60s and 70s, which will tend to prolong their life for reasons of compatibility, cost of re-training and so on. This multiplicity of languages, coupled with a recognition of the need to move towards re-usable software, argues for multi-language IPSEs where systems can be built out of components in a variety of languages.

Similar considerations apply to mixed hardware and software systems. It is clear that there are enough similarities between the hardware and software design processes, and the administrative and management procedures appropriate to them, for there to be benefit in using one IPSE for hardware and software development. Furthermore, it is important that the requirements analysis, functional specification and much of the design work can be done independently of decisions whether particular modules should be implemented in hardware or software. For such modules, their function must be defined, their place in the overall design established and their performance requirements known; economic, timescale and other criteria may then be used to determine how they should be implemented.

A fully integrated IPSE as just described is exactly the Information System Factory that is the major objective of the programme. It is a long term objective, but it is important to be clear about what the objectives are in order to see how to move towards them, and in particular to determine the role of UNIX and Ada APSE developments in this process.

One conclusion that emerges strongly is that there is still a great deal of research and development to be done before such an integrated PSE can be built. Two important areas needing R & D are:

i. formal, rigorous methods of specification of requirements, and techniques to express designs and determine how far they meet their specifications for performance, reliability, correctness etc;

ii. methods of structuring software or hardware system components for wide re-use; the nature of their interfaces to each other, the appropriate types of global design to incorporate them; how to document them; how to search for and locate them.

The Alvey programme will include one or more evolving IPSEs, which not only bring together existing tools and procedures to improve development cost-effectiveness in the shorter term but are also capable of incorporating new techniques that emerge from relevant R & D projects.

3.3.1 ISF and the Three IPSE Generations

The Integrated Project Support Environment (IPSE) is a major product objective of the programme. The programme will proceed as follows.

(1) Commission development and creation of three generations of IPSE:

1st)) file

2nd) generation IPSE) database

3rd)) knowledge base

(2) Versions of each generation of IPSE to be sited in SPC (section 3.5) and NQCC (section 3.6) and selected organisations where IPSE impact on quality and productivity can be monitored and reported.

(3) Cooperate with and incorporate aspects of other Alvey areas towards ISF eg CAD for VLSI, high resolution displays, expert systems for programmers.

3.3.2 The 1st Generation IPSE

UNIX will be used as the basis for the 1st generation (file based) IPSE. UNIX is rapidly becoming a de facto standard over a very wide range of systems and organisations and therefore offers the prospect that:

- There will be many developments for UNIX which can be taken advantage of by the Alvey programme.

- The market for UNIX-based development environments and tools is large and growing.

These factors should minimise the amount of tool integration and development needed to improve today's UNIX environment into a genuine 1st generation IPSE.

This is not to say that the Alvey programme is endorsing UNIX as a standard; UNIX will be used as a starting place. Nonetheless, it is envisaged that an active UNIX community will come into being in the early years of the programme, supported by communications network facilities.

3.3.3 The 2nd Generation IPSE

The second generation IPSE contains two major components not found in the 1st generation IPSE:

(1) Database-based tool set (rather than file-based) eg CADES.
(2) Support for geographically distributed project teams e.g. Newcastle Connection.

As (1) and (2) above are somewhat orthogonal it is expected that several approaches will be attempted, including the evolutionary development of the 1st generation, UNIX-based IPSE as well as the 'clean sheet', non UNIX attack, possibly via intermediate steps which will contain one, but not both, of the distribution and database components.

The 2nd generation IPSE software will run on new hardware; developments in cheaper CPU power, cheaper, high resolution colour graphics, and non keyboard input-output devices, for instance, will facilitate productivity gains due to improved man-machine interaction. The 2nd (& 3rd) generation IPSE will require new hardware based components such as:

1. Single user workstation costing £5K with A3 black and white 2K x 2K pixel graphics.

2. Single user workstation costing £10-20K with A3 colour 2K x 2K pixel screen

3. 100 Mbits/sec local area network.

4. Gateway to high speed (greater than 1 Mbit/sec) wide area communications.

5. LAN servers for files and databases.

6. High quality, cheap print server eg. laser printer.

7. Full-generality distributed operating system.

8. Sophisticated man-machine interface.

The Programme will stimulate the UK production of hardware suitable for the 2nd Generation IPSE as described above. It is important to effect this development in a short enough timescale to prevent UK manufacturers being eclipsed by the USA and Japanese industries in this large market.

3.3.4 The 3rd Generation IPSE

The 3rd generation IPSE (or ISF), containing knowledge bases and 'intelligent' tools, requires significant research which must begin now if the 1989 target date for the Information System Factory is to be met.

It is envisaged that the ISF will be defined as much by market and economic realities as by any technical goals; it will (almost by definition) embody the most cost-effective ways of producing application-specific IT systems available at the time. An Information System Factory will probably consist of six main subsystems:

1. specification and prototyping facilities

2. a Software Development Environment

3. a facility for CAD of VLSI and hardware development

4. a database or knowledge base of available, reusable software and hardware components

5. the communication systems, both local and wide area, to facilitate co-operative development

6. project management aids.

How far advanced these six subsystems are by 1989, and how closely integrated together, depends on technical advances which are hard to predict. Markets will exist for the separate components as well as the unified ISF. The following sketches are probably optimistic in their assumed rate of technical progress, but help to define the aims.

1. Specification and Prototyping Facilities

Specifications of the system under development will be held internally in a formal, machine-manipulable form (which is central to the integration of the whole ISF - since it is used by all its subsystems). There will be extensive facilities to convey these specifications to people such as system designers and the eventual users of the application system - by animating the specifications, producing small prototype systems, question-answering and so on. Completeness and consistency of the specifications will be checked automatically.

The nature and role of prototypes is often misunderstood and confusingly. Prototyping is a well understood technique in other areas of engineering. A prototype should only be constructed purposefully to answer a (set of) specific question(s) defined in advance of its construction. Prototypes may be more or less expensive/sophisticated/realistic, when compared to the production version, depending on the nature of the questions asked of them (contrast 'breadboard' v VLSI chip, wind tunnel model v Concorde).

2. Software Development Environment

This will go beyond present-day environments in supporting all phases of the software lifecycle and in relating them back to the formal specifications. It will be tailored to support one of several different development methodologies - depending on the application area - and will support a defined style of project management.

3. Facility for CAD of VLSI and Hardware Development

With the emergence of special-purpose hardware architectures implemented in VLSI, the need arises for functions to migrate between software and hardware during the lifetime of an application system. So CAD of VLSI cannot be considered as a separate problem. A VLSI-implemented system must meet the same formal specifications as a software system, and pass the same tests, and vice versa. Therefore the software development and the VLSI CAD facility need to be centred around the same specification method and must communicate with one another.

4. Database of Available, Reusable Components

To compete effectively in making IT application systems, it will be increasingly necessary to re-use existing software and hardware components. These components will be very diverse - a component could be a software product, an integrated circuit, a sub-routine or fragment of code, an algorithm, a man-machine interface device or one of a set of formal theories about data structures. A database of such components will hold some information common to all of them, to answer questions such as: What does it do? (i.e. its formal specification) What environment does it require? (language, storage space, power requirements, inter-connections etc) and Can it be adapted to perform a slightly different function? Some components will be general purpose and some will be application specific. Initially such information will be held in a database and searched in various ways; but in the longer term there is a need for automatic reasoning based on the data; this broadens the requirement to an intelligent knowledge based system (IKBS).

5. Communications Systems

A key feature of the ISF is the facility to allow groups of designers and programmers to work co-operatively, even when geographically distributed. This will mean a requirement for high bandwidth communication between co-operating processes, both within site, and between sites. It must be possible to implement the ISF in a distributed manner. It is expected that whilst the basic communications technologies of local and wide area networks will exist to allow this to occur, nevertheless considerable developments will be required to meet the special needs of the ISF, expecially in handling interactive high resolution colour graphics.

6. Project Management Aids

Project planning, management and control methods will be developed. When supported by a comprehensive collection of tools, these management techniques will provide both professional managers and technical staff with the ability to effectively plan and control all aspects of the software development process throughout the life cycle. These management tools must be intimately integrated into the development process to ensure that all the appropriate parameters can be realistically measured.

Thus an ISF, with all six subsystems implemented to a greater or lesser extent, will be an essential prerequisite to compete in producing medium to large scale information systems in the late nineteen eighties. It will represent a major capital investment for anyone intending to compete in the field.

The discussion so far has concentrated on the development of large, complex application systems; however, similar remarks apply equally to the small systems market. To remain competitive in producing IT products, companies will have to use advanced specification and prototyping tools, application development aids and libraries of components to produce better systems faster. So analogous small-scale Information System Factories may well dominate the small systems field, although market forces will drive them more to a low cost, high volume regime. The greater dynamism and adaptabiliy of this sector development and pure research. The theoretical underpinnings of software engineering are considered to be of vital importance - a thorough 'understanding' must precede the expensive construction of the sophisticated ISF-type environments.

3.5 National Quality Certification Centre

The primary medium term payback activity is seen as the creation of a National Quality Certification Centre (NQCC) for software products and components. The NQCC must build up an international reputation. This will involve the adoption of state of the art techniques on a continuous basis. The commercial benefit of NQCC approved software products in an international market is potentially extremely valuable. As the mass market for software products develops consumers will buy NQCC approved products rather than unapproved products. The rapid establishment of such a national capability could give the UK a significant commercial advantage.

The concepts behind the NQCC are currently in their infancy with only communications protocols and programming language compilers being 'certified'. The NAG library quality control reputation shows the potential benefit of extending this concept.

The NQCC should provide a realistic focus for much speculative research and development work.

The NQCC cannot spring into existence overnight. It is envisaged that early in the programme one or more R & D centres will be established to develop quality assurance and certification techniques. At least one centre's medium term aim will be to transform itself into the Alvey Quality Certification Centre. If the AQCC can establish a national reputation then the move to genuine 'national institution' status, possibly as an independent, revenue earning body, should rapidly follow.

3.6 Software Production Centre

The SE programme will establish a Software Production Centre. This will not be a research project but a working factory funded to exploit and incorporate the latest technology. The facilities of the centre will be made available to software producers for genuine production work. This will enable large organisations to try out 'real' new techniques before making the necessary in-house investment. It will also enable small companies to experience the benefits of new technology which they could otherwise never afford.

The SPC will be aimed at producing software which will pass the tests laid down by the National Quality Certification Centre.

Technically, the SPC is to be a multi-lingual, database foundation, integrated project support environment. It will act as the focus for the practical embodiment of much research and development work. It will support not only the development of new software but the maintenance and evolutionary development of existing products. To this end, it will be 'multi-lingual', ie it will be capable of developing systems in, say, Cobol, Fortran and Coral as well as eventually, say, Ada and Prolog. It will also be 'multi-lingual' in the sense that any one of its software products can be constructed from components coded in several different languages. Such a requirement will stimulate the development of re-usable software components and maximise the return on investment in existing software.

The technology contained within the Software Production Centre could be exported into the sites of the participating organisations by:

a. replication of hardware and software components on the site

b. network access from the site to the Centre

c. a combination of a and b.

3.7 Development Programme

The NQCC and Software Production Centre require a research and development programme to feed them with new technology. The Director (SE) will initiate a medium term R & D programme to ensure that the goals given in section 1 are realised throughout the UK. This will require a balanced programme of directed contract work and responsive funding.

3.8 Software Reusable Components Brokerage

The Director (SE) will examine the desirability and feasibility of a centre for software components and products. Software components include more than just modules of source code or relocatable binary. Components include documentation, design documents, forma specifications, test data sets, test results, benchmarks, and cost estiumation models, PERT charts and other project management items.

It will operate by holding specifications, code etc in a database accessible only via the Alvey network. Dissemination of components will be only by FTP (file transfer). Participants will lodge their components and products in the database with distribution at a charge. This scheme should encourage collaboration, technology transfer and the creation of reusable software components and products, the idea being that it will be quicker and cheaper to get a subroutine from the brokerage than to reinvent it.

Two types of software products are envisaged as being handled by the Brokerage:

a. Packages for sale to the public. These should ideally have been approved by the NQCC.

b. Reusable software components. These too should ideally have NQCC approval but will not be on sale generally. They will be available to the 'trade', ie to those software developers who will attain increased productivity by using existing components rather than developing their own and who will contribute components of their own manufacture. The Software Production Centre should be a major source of, and customer for, these components.

3.10 Market Sectors Addressed by the SE Programme

The Alvey SE Programme is aiming to be of relevance to all large scale, professional software producers, including the Data Processing sector. The distinction in methods, tools and practices which have existed between say, the real time military and industrial sector and the DP sector is regarded as diminishing and will vanish in the 1990s as the IPSE and ISF concepts are realised.

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

strat/seg 113.2