The Benefits of CASE: Myths and Reality

BUTLER COX
P,E,P

PEP Paper 20, November 1991

BUTLER COX

P.E.P

The Benefits of CASE: Myths and Reality

PEP Paper 20, November 1991
by Kevan Jones

Kevan Jones Kevan Jones is a consultant with Butler Cox, working both in London and Amsterdam. He specialises in systems development. He has been involved in PEP since its inception, carrying out research into the systems issues that are of concern to members, analysing members' systems development productivity, and more recently, managing the PEP service in the Benelux countries. He is the author of PEP 10, *Making Effective Use of Modern Development Tools*, the Butler Cox report, *Improving Systems Development Productivity and Quality*, and the Butler Cox Foundation Report, *The Future of System Development Tools*.

During his time with Butler Cox, he has also participated in a wide range of consulting assignments. Recent projects in which he has been involved include assessments of the effectiveness and efficiency of the systems development departments of several large clients, and an assessment of the process-control systems of a computer-integrated manufacturing installation for a major oil company.

Prior to joining Butler Cox, Kevan Jones was a member of the electro-optics systems team at Thorn EMI Electronics, where he worked on systems analysis, design and implementation of large projects for the Ministry of Defence.

He has a first-class honours degree in mathematics and computing and is an Associate Fellow of the Institute of Mathematics and its Applications.

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

Printed in Great Britain by Flexiprint Ltd., Lancing, Sussex.

BUTLER COX

P.E.P

The Benefits of CASE: Myths and Reality

PEP Paper 20, November 1991
by Kevan Jones

Contents

. . . continued

Contents

Chapter 1

CASE is popular, but not a success

Development departments are under increasing pressure to improve both the rate of delivery and the quality of the applications that they produce. Computer-aided software engineering (CASE) tools promise to meet this shortfall in development productivity and quality. The benefits claimed by CASE suppliers range from ten-fold productivity improvements, to applications developed with zero errors, and almost every claim is accompanied by examples of how various organisations have achieved these benefits.

For most organisations, the benefits of CASE are proving elusive

Adopting CASE is, however, an expensive undertaking and organisations therefore need to be sure that the scale of the benefits will justify the large and long-term investment that will be required. Very few organisations are, in fact, achieving significant benefits; for most, the promises are proving very elusive and the obstacles insuperable. According to recent reports in the press, some are currently reassessing their decision to adopt CASE, or are abandoning it altogether (see Figure 1.1).

Figure 1.1 The potential problems with CASE are beginning to be realised

Dun & Bradstreet, one of the world's largest software providers, recently completed a year-long evaluation of over 100 CASE products and failed to find any robust enough to build future packages. It concluded that the major weaknesses of CASE were its inability to achieve cross-platform consistency, insufficient improvement in software quality and inadequate speed-up in development time.
BP Oil recently replaced Information Engineering Workbench (IEW) from KnowledgeWare with PREDICT-CASE from Software AG. The deal is alleged to be worth £500,000.
British Gas, which, as a corporation, is understood to have made a strategic commitment to IEF in one of the biggest CASE deals signed to date, is now reviewing the situation. British Gas North Thames and South Eastern undertook a complete rewrite of their corporate systems. It is now believed that the projects were too ambitious and that they have been hampered by shortcomings of IEF, such as problems with version control.
American Express, the US financial services company, is reviewing its use of IEF. American Express has been using IEF since 1988 and it is currently being used on a multimillion-dollar redesign of its computer systems. The company is now questioning how appropriate IEF is to the tasks being carried out.

Our research shows that most organisations are not yet in a position to exploit CASE tools fully – especially the front-end planning and design tools. We strongly recommend that PEP members do not adopt CASE until they are able to show, by assessing their readiness, that the benefits will accrue. Until then, their limited resources are better spent elsewhere within the development department.

Good management is required to get the best from CASE

For those that are ready, CASE need not be a disappointing and wasteful experiment. To get the most out of what is undoubtedly an expensive undertaking, however, managers will have to be prepared to be rigorous and uncompromising in their approach to CASE and in the way they continue to monitor and support it once it is in place.

CASE tools were first introduced more than 15 years ago, and since then, the term has been extended to include many different types of tools. Those classified as CASE for the purposes of this paper are listed in Figure 1.2. There are CASE tools that cross the boundaries of, and lie outside, this classification – for example, reverse-engineering tools, which are specialised support tools for the CASE environment that require special attention and different initiatives, are excluded. We believe, however, that it covers virtually all the CASE tools used by PEP members today.

Figure 1.2 Many types of tools are included in the term CASE

Type of tool	Description	Examples
Strategy planning tools	Provide computerised support for the planning and high-level control of the development resources	Strategic Planner TETRARCH 1 Prism
Front-end tools	Provide computerised support for the analysis, design and documentation of applications	Excelerator Auto-Mate Plus DesignAid
Back-end tools	Provide computerised support for the construction of applications	APS Telon Delta
Life-cycle tools	Provide computerised support in an integrated manner across several stages of the development life cycle	IEW PACBASE CASE*
Project-management tools	Provide computerised support for the management of projects	PMW CA-SuperProject Wings

Despite the high costs, CASE is proving very popular

Most CASE tools are expensive

One of the few facts that is proven about CASE in general is that the costs are high. Although there are some PC-based CASE tools that are relatively inexpensive – for instance, the front-end tool, EasyCASE Plus, from The Software Construction Company, which costs £190 – most cost about £4,000 per user. The cheaper tools are aimed at standalone PC development and tend to lack the facilities required by large development departments. Generale Bank, one of the largest banks in Belgium, has spent over £500,000 adopting APS, the code generator from Intersolv, and IEW, the Information Engineering Workbench from KnowledgeWare, during the past 18 months – an average cost of nearly £2,000 per developer per year. Similar costs have been reported in the United States, with an average cost per developer of £4,000 per year for a development department with 200 staff. One consulting firm is telling major IBM clients that, with a full CASE environment, it will cost

$10 million to write their first MVS repository-based application ($4 million on software and hardware and $6 million on people).

Nevertheless, many organisations, tempted by the promises of major gains in productivity and quality, have adopted CASE. Most PEP members (88 per cent of the respondents to the survey for this paper) make some use of CASE, although the percentage of the development workload supported by CASE is quite low (27 per cent). This trend is confirmed by a survey of 300 organisations in the United States; 80 per cent were using CASE, but only 10 per cent of the workload was supported by it.

There is an enormous and increasing range of CASE products available

The popularity of the tools is clearly reflected in CASE sales worldwide, which have increased nearly four-fold since 1987, and were estimated at $750 million in 1990. With such a large market, it is not surprising that there is an enormous range of products on offer (over 1,000 according to a recent survey) and that new ones are constantly appearing – at peak, about 100 announcements of new CASE products/versions each month.

Most organisations are not realising the promised benefits

Our research has shown that high levels of benefit are *not* being achieved by the majority of organisations. We surveyed PEP members to compare the levels of benefit that they expected to achieve and what they actually realised with CASE. For 70 per cent of the members, the levels of benefit achieved with CASE were lower than expected, in all areas.

Detailed analysis across the whole membership showed that benefits exceeded expectations in only one area – upgrading the skills and knowledge of systems staff. CASE had not provided benefits above the expected levels in terms of reduced costs, speedier delivery, or reduced maintenance effort – all areas in which CASE suppliers and experts claim that great savings can be made. The detailed results of our questionnaire on benefits are shown overleaf, in Figure 1.3. Analysis of the PEP database, discussed in detail in Chapter 2, verified these results and quantifiably demonstrated that the use of front-end CASE tools typically results in a *decrease* in productivity and quality.

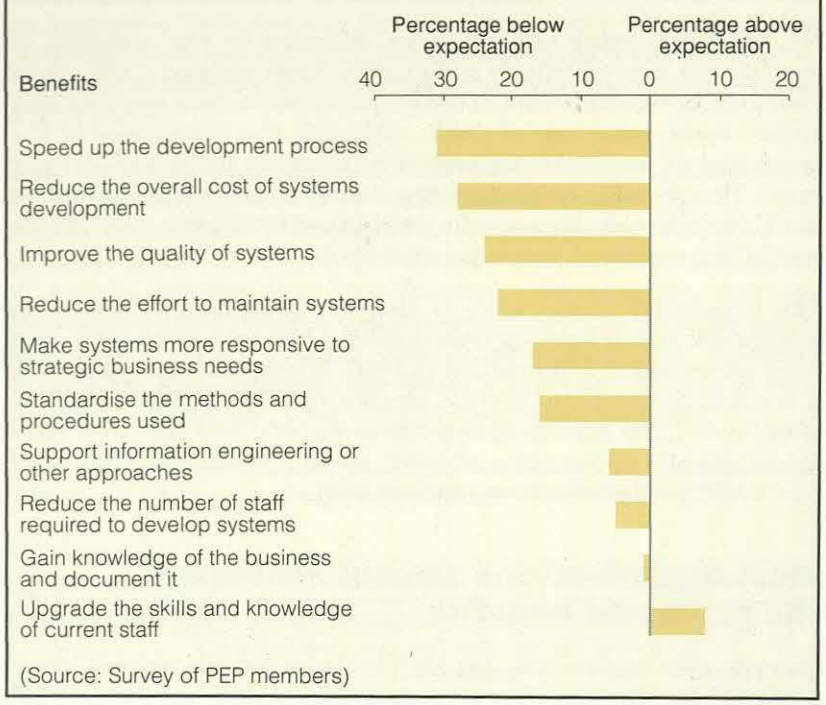
The use of CASE tools often results in a decrease in productivity and quality

The disappointed expectations of our respondents can be explained by the fact that CASE benefits are being oversold and that PEP members are still encountering a whole range of problems with CASE. Five of the problems most frequently quoted during our research are discussed below. They are similar to those reported by the CASE Research Corporation in a survey carried out in late 1989 of 270 medium-to-large IBM mainframe-based organisations. The full results of this part of the survey are shown in Figure 1.4, also overleaf.

PEP members are facing many problems with CASE

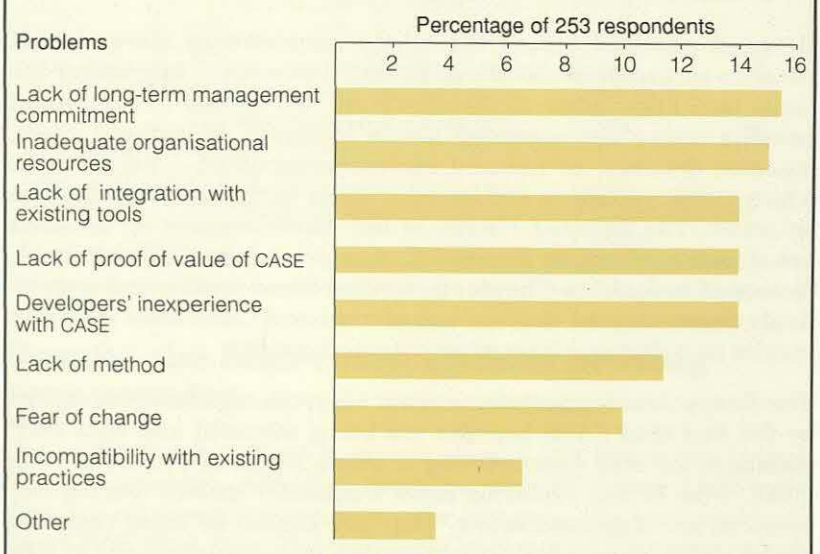
Lack of long-term commitment. CASE tools require long-term commitment for any real benefits to be achieved. At the heart of most CASE tools is some form of database in which the information used by the tool is stored. These databases are known as data dictionaries or repositories. Before many CASE tools yield noticeable benefits, these dictionaries needed to contain a reasonable amount of information. As the use of CASE increases, the in-

Figure 1.3 The perceived levels of benefit associated with CASE are much lower than expected



(Source: Survey of PEP members)

Figure 1.4 There are still many problems with CASE



(Source: CASE Research Corporation)

formation in the dictionary becomes increasingly critical to the maintenance of the applications. Thus, in deciding to move to CASE, the organisation is committing to the dictionary, in the long term, to support the maintenance of its applications.

One organisation had difficulty maintaining commitment for CASE, and in 1989, stopped generating applications with IEF from James Martin Associates (now JMA Information Engineering) after

investing approximately £300,000 in the product over the previous one-and-a-half years. Having had difficulty using the product on several developments, it decided to use IEF only for the initial planning stages of a development project. Both the company and JMA blamed many factors, including a lack of management commitment. The technical development manager said that while he had no argument about the tool itself, the development department wished, in retrospect, that it had not gone down this particular path. It had underestimated the costs, the impact and the need for consultancy and training.

Inadequate organisational resources. Many organisations fail to provide the resources and expertise required to support the use of CASE tools within their development department. Experts will be needed to support the technology, which includes the hardware, the software and the method. Experts, such as database administrators and controllers, will be needed to control the use of CASE, and the whole CASE environment will need to be managed and developed. Without such organisational support, the use of CASE may either be restricted or work may be duplicated by other staff, thus creating extra work to consolidate it. CCHA Housing, a UK-based housing association, believes that it made a mistake in not appointing a database administrator at the outset. It used Synon; each modeller constructed his own model, all of which subsequently had to be combined, which meant that many duplications had to be sorted out.

Lack of integration with existing tools. Although many of the tools in the CASE environment are integrated in some manner, there is little support for linking CASE tools to existing tools such as fourth-generation languages and certain third-generation languages. This may mean that the use of CASE tools is limited to new developments or to a particular area of development within the business. Re-engineering tools promise to help in this area by automatically translating old applications, developed with existing tools, into the CASE environment. Various forms of re-engineering tools are gradually being adopted, but they are not fully automated and they require a considerable amount of human intervention. Re-engineering is still a long way from widespread acceptance.

CASE is a benefit multiplier, not a benefit generator

Lack of proof of the value of CASE. Despite the impressive claims made for CASE, the benefits have been difficult to prove in many organisations, partly because appropriate measurement programmes have not been in place to monitor what happens, and partly because CASE is primarily a benefit multiplier rather than a benefit generator. By this, we mean that CASE typically automates, or semi-automates, parts of the development process – it improves, in some manner, the efficiency of the process that is already in place. If the development process results in poor applications, CASE will not improve the situation; at best, it will add some formality. This was confirmed by development managers in both the United Kingdom and the United States. The productivity manager at ICI Chemicals and Polymers in the United Kingdom said, "If you want to produce benefits from CASE, you must do so; it [CASE] will not produce them". Tim Lister, of the Atlantic Systems Guild, said at the CASE World conference in Los Angeles, "Software [development] is hard and CASE won't make it easy. If you're good at building software, CASE tools will help. Otherwise, they won't."

Developers' inexperience with CASE. Inexperienced users of CASE can produce systems that are supported by such complex data structures that each new system becomes more difficult to deliver. It is also possible for analysts, designer and developers to spend a considerable amount of time and effort using the CASE tool and adding no value to the delivered application. Early experience with fourth-generation languages was quite similar. There were claims of enormous benefits, but it became apparent that inexperienced developers, without proper direction, support and training, were simply likely to develop a disaster faster.

CASE requires rigorous analysis and supporting initiatives

We believe that the problems discussed above are not insurmountable, and for those who are ready for CASE, it can provide an environment in which the promised productivity and quality benefits can be realised. Indeed, some organisations have reported high levels of success. These organisations are involved in quite varied lines of business and do not necessarily have large development departments working on large-scale commercial systems. For example:

The promised productivity and quality benefits can be realised by those ready for CASE

- CCHA Housing has a development department consisting of only six analyst/programmers. It currently uses KnowledgeWare's IEW, Computer Systems Advisers' Picture Oriented Software Engineering (POSE), and Synon's code generator. Although, like many other organisations adopting CASE, it has had its problems, it believes that it is now developing systems that are 'more solid' with far fewer errors, although no hard evidence was available to support this. The business systems manager also stated that, "As a not-for-profit organisation, we cannot offer huge salaries, but we can compete in offering an interesting working environment, with a chance to work with CASE and fourth-generation languages".

- Caisse Régionale du Crédit Agricole Mutuel de l'Ile de France, a regional bank with 250 branches in France, attributed an 80 per cent decrease in failures (once the system went into full use) to the CASE tool, PACBASE, from CGI Systems and to its own quality-control approach. Less effort was also required to fix the incidents. In the last three years, the amount of effort spent on maintenance has fallen from 500 to 100 man-hours per month.

- Yorkshire Water, one of the privatised UK water utilities, uses IEW and an application generator, CorVision, from Cortex. The organisation believes that it is getting a six-fold improvement in productivity. It is delivering three times the function, measured in function points, in half the planned time.

Clearly, improvements are being achieved with CASE in some development departments. However, as we have already shown,

A systematic approach will remove some of the risk from adopting CASE

rigorous management is required, and the costs are high. Organisations need to understand, assess and weigh up these high costs and the benefits associated with CASE. In this paper, we suggest a systematic approach that will be a useful guide to PEP members in what is, undoubtedly, a risky undertaking.

In Chapter 2, we show the quantifiable figures behind the often-quoted costs and benefits associated with CASE tools. Clearly, the costs are high, and in many cases, the benefits very limited.

In Chapter 3, we discuss how an organisation should assess its real needs and its readiness to move to, and to continue to operate in, a CASE environment. Many organisations have failed with CASE because they have adopted it prematurely and eventually had to discard it.

In Chapter 4, we examine the management initiatives taken by those organisations achieving significant benefits from CASE. Only by adopting the appropriate initiatives to support CASE will the high level of benefits currently being achieved by the minority be realised by the majority.

In Chapter 5, we argue for the establishment of a measurement scheme so that it is clear what the costs of CASE are, where benefits are being achieved, and where there is room for further improvement.

Research sources

The prime source of the information on which this paper is based is data provided by PEP members. We circulated a questionnaire to all PEP members and received 80 completed questionnaires from 75 organisations. We subsequently discussed particular issues in more detail with 15 members.

As of August 1991, the PEP database contained 862 projects. We categorised these projects according to the types of CASE tool used, and analysed and compared the results of those projects with the results of non-CASE projects to demonstrate the average level of benefit currently being achieved.

We also carried out an extensive review of the documented histories of CASE users and the current best advice and practice with CASE. (A short bibliography is included at the end of this paper.) As a result of this investigation, we conducted several detailed interviews with organisations that are not PEP members.

We believe that the CASE marketplace is starting to go through a period of rapid change, with many of the suppliers either merging or being bought out. We therefore reviewed the suppliers of the more popular CASE products to provide members with an indication of their ability to support differing needs in the future. The results of the review are recorded in Appendix A.

Chapter 2

For most, the benefits of CASE are proving elusive

The concept of CASE is reasonably well understood within the industry and most development managers believe that they also understand how CASE will affect their environment. In fact, very few do. Our research has shown that, on average, the benefits achieved with CASE do not justify the costs. Yet organisations continue to invest in CASE.

Although we were unable to quantify all the benefits of CASE tools, we were able to show how CASE is affecting development productivity and the technical quality of delivered systems. Figure 2.1 gives an overview of the benefits currently being achieved. It shows that front-end CASE tools generally have a negative impact. There may be a slight positive impact on maintenance and on improving the fit with business needs, although we have not been able to measure this directly.

Front-end CASE tools have a negative impact on development productivity

Few quantified improvements can be demonstrated

At the time of writing, the PEP database contained 320 projects on which the use of CASE was reported. We compared these with the 507 projects on which there was no reported use of CASE tools. This analysis revealed that, overall, CASE tools were having a small negative impact on function delivered per man-month and a small beneficial impact on the technical quality of delivered systems. The results of this analysis are summarised in Figure 2.2, on page 10.

Overall, projects on which CASE tools are used produce nearly 10 per cent fewer function points per man-month of effort. The time taken to produce a function point is less than that for projects on which CASE was not used, but both the time and effort will have been affected slightly by the higher than normal time pressure on the projects. Increased time pressure typically results in an increased level of effort and reduced delivery time for a given size of system.

CASE projects produce fewer function points per man-month than non-CASE projects

The CASE projects were typically larger, measured both in function points and lines of code, than projects on which CASE tools were not used. This is probably due to the limited use of CASE tools on small maintenance projects. Although there is no significant shift in the proportions of time spent on the main-build and functional-design stages, the data shown in Figure 2.2 implies that there is an increase in effort in the functional-design stage of nearly 25 per cent. The implication is that the use of front-end CASE tools increases the amount of effort required to analyse and design the system.

Figure 2.1 The benefits being achieved with most CASE tools are very limited

Benefit	Analyst workbench	Programmer workbench	Code generator	CASE overall
Reduce development cost	– –	– –	++	–
Reduce development time	– –	+	++	+
Improve technical quality	– –	+/–	++	+
Improve reliability	– –	–	++	+
Reduce maintenance workload	+	+/–	+/–	+/–
Reduce maintenance costs	+/–	+	+	+/–
Reduce maintenance time	+/–	+	+	+/–
Improve fit with business needs	+	+/–	+/–	+/–

– – Significant negative impact

– Negative impact

+/– No evidence to suggest positive or negative impact

++ Significant positive impact

\+ Positive impact

Impact quantified through analysis of PEP database

Impact assessed through discussions with CASE experts and PEP members

(Source: PEP database)

CASE has a small positive impact on the quality of the delivered system

Use of CASE appears to have a positive impact on the technical quality and reliability of the delivered system. During both systems testing and the first month of operation, an average of 10 per cent fewer errors were detected.

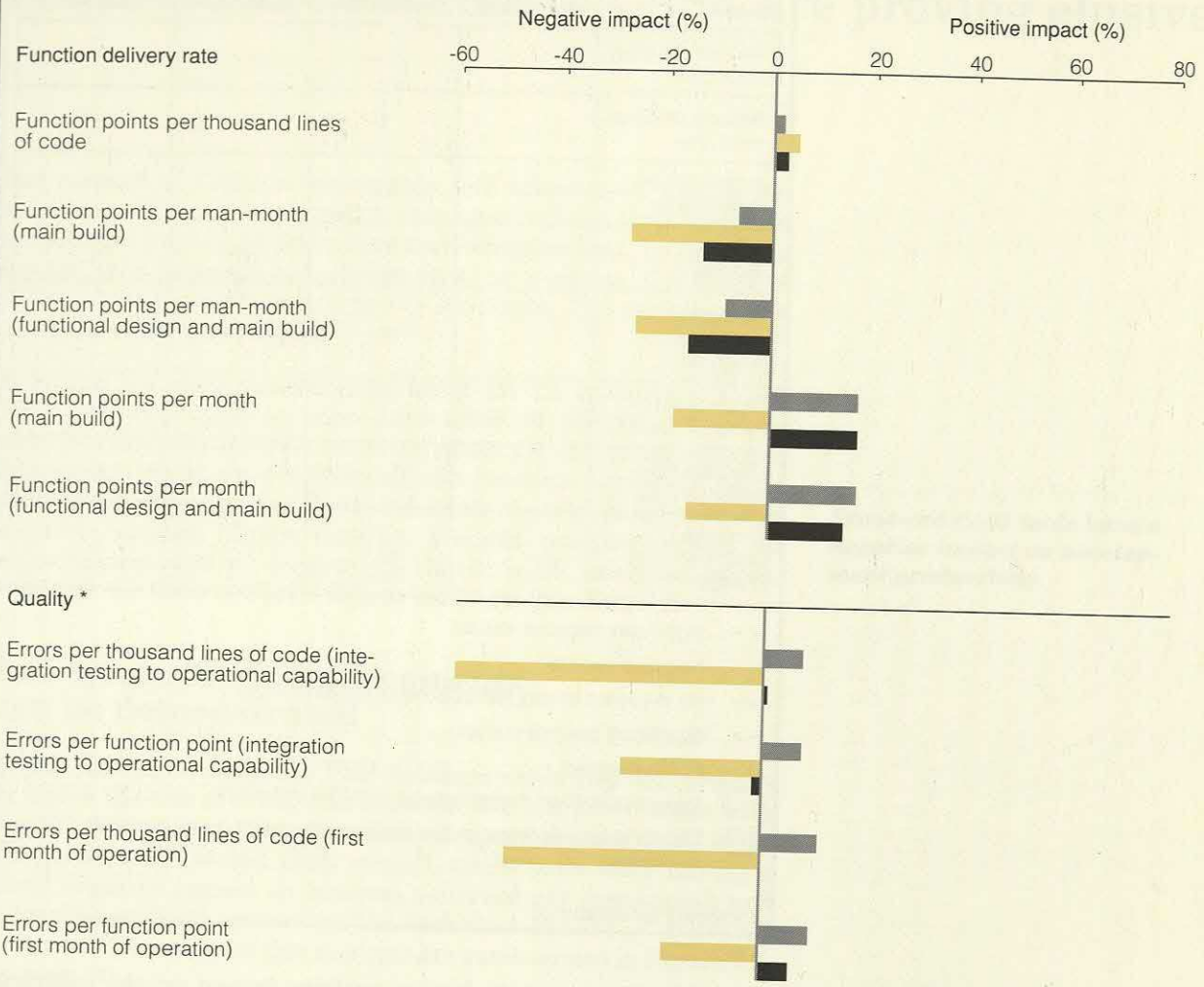
We are unable to comment on the impact of CASE on improving a system's fit with users' needs or on reducing maintenance. However, our survey of PEP members revealed that benefits were much lower than expected both in reducing maintenance effort and in delivering better-quality systems. CASE was also rated very low in terms of making the department more responsive to strategic business needs.

Analyst and programmer workbenches offer mixed results

The PEP database contained 36 projects on which analyst workbenches were used. Our initial analysis of these projects produced contradictory results; some were performing much better than others. Further investigation showed that the analyst workbench

Figure 2.2 Few quantified benefits can be demonstrated for front-end CASE tools

Projects in the PEP database reported as using CASE tools, analyst workbenches and programmer workbenches were compared with 507 projects where no use of CASE was reported, to assess the impact of these tools on function delivery rates and quality.



320 projects developed with CASE tools; average size 479 function points or 37,000 lines of code (about one-third larger than non-CASE projects).

8 projects developed with analyst workbenches; average size 420 function points or 32,000 lines of code (both slightly larger than non-CASE projects).

140 projects developed with programmer workbenches; average size 526 function points (50 per cent larger than non-CASE projects) or 40,000 lines of code (45 per cent larger than non-CASE projects).

* Error data was available only for 215 CASE-tool projects, 4 analyst workbench projects and 86 programmer workbench projects.

(Source: PEP database)

projects that performed better also used fourth-generation languages. We therefore analysed these projects separately, leaving just eight projects that used only analyst workbenches. The analysis of these eight projects is also shown in Figure 2.2. For these projects, the number of function points delivered per man-month is about 25 per cent lower than for projects on which CASE tools were not used. Although it can be argued that analyst workbenches are not expected to increase productivity levels, it is

Analyst workbenches have a negative impact on function-point delivery rates

important to note that they are having such a dramatic negative impact in terms of function-point delivery rates.

We are not able to comment confidently on the technical quality and reliability of analyst workbench projects that did not use fourth-generation languages because errors were reported for only four of them. However, the limited data available suggests that the use of analyst workbenches has a significant negative impact on technical quality and reliability (see Figure 2.2 again).

These results confirm the negative impact of analyst workbenches previously reported in PEP Paper 12, *Trends in Systems Development Among PEP Members*. Analyst workbenches may, of course, help to produce systems that are a better fit with business needs, but the PEP database does not at present enable us to measure or demonstrate such improvements.

Programmer workbenches result in lower-than-average productivity

Figure 2.2 also shows that the 140 programmer workbench projects in the PEP database produce fewer function points per man-month, even though the function points per thousand lines of code is close to that achieved on projects that did not use CASE tools. Thus, programmer workbenches result in lower-than-average productivity. This is primarily due to the nature of the programmer workbench projects – 45 per cent larger than normal, written in traditional languages and developed under slight time pressure, which results in higher-than-average function points per month. These results confirm our earlier analysis for PEP Paper 12 of the impact of programmer workbenches on productivity and it is disappointing to find that the situation has not improved.

The technical quality of projects that used programmer workbenches is close to the average for projects that did not use CASE tools. Although this is an improvement since the research for PEP Paper 12, it is still disappointing. Overall, our analysis shows that programmer workbenches do not dramatically improve the delivery rate, technical quality or reliability of the systems delivered.

Code generators improve both productivity and quality

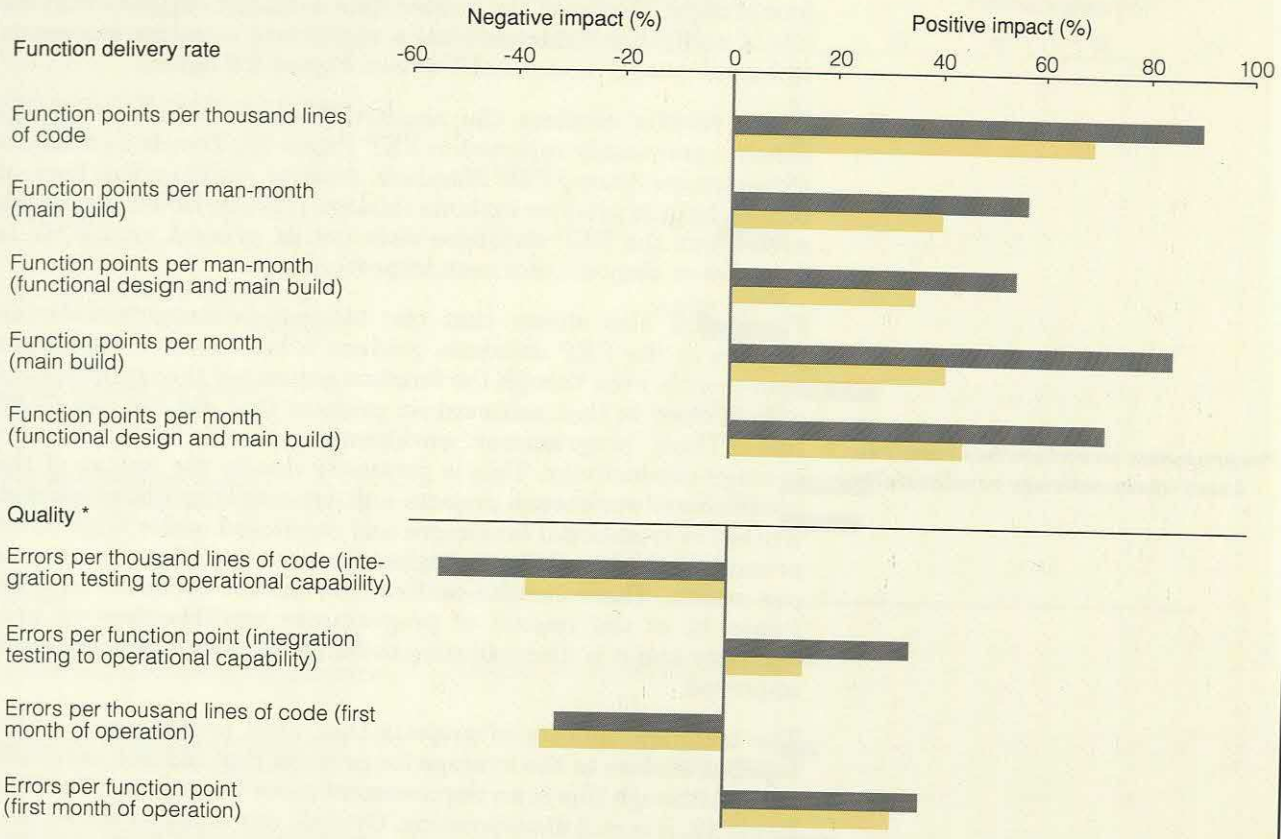
Code-generator projects are much more productive than non-CASE projects

Figure 2.3, overleaf, shows clearly that projects on which code generators are used are much more productive during the development process than projects that do not use CASE tools – typically producing 55 per cent or more function per month and per man-month. Although the size of the systems, measured in lines of code, is similar to that of non-CASE projects, the high language gearing (function points per thousand lines of code) results in over 90 per cent more function being delivered.

The quality of these systems is also much better in terms of errors per function point. On average, there are 35 per cent fewer errors per function point, both during testing and once the system is in use – that is, systems are being developed with one-third fewer errors. The increased number of errors per line of code, however, indicates that there are still errors being introduced either by poorly trained staff or by programmers who misunderstand the specifications or who make mistakes as they link new modules to one another or to existing modules.

Figure 2.3 Code generators and fourth-generation languages provide significant benefits

Projects in the PEP database reported as using code generators, and analyst workbenches with fourth-generation languages, were compared with 507 projects where no use of CASE was reported, to assess the impact of these tools on function delivery rates and quality. Most of the improvement with analyst workbench projects is due to the use of fourth-generation languages.



30 projects developed with code generators; average size 681 function points (94 per cent larger than non-CASE projects) or 28,000 lines of code (close to the non-CASE average).

28 of the analyst workbench projects that also used fourth-generation languages; average size 600 function points (71 per cent larger than non-CASE projects) or 28,000 lines of code (close to the non-CASE average).

* Error data was available only for 25 analyst workbench/fourth-generation-language projects and for 14 code-generator projects.

(Source: PEP database)

For comparative purposes, Figure 2.3 also shows the impact of the 28 analyst workbench projects that used fourth-generation languages as well. By comparing the analysis of these projects with the other eight analyst workbench projects analysed in Figure 2.2, it appears that fourth-generation languages typically increase productivity by up to 60 per cent. Thus, the use of fourth-generation languages results in improvements similar to those achieved by using code generators.

Fourth-generation languages typically increase productivity by up to 60 per cent

CASE costs currently outweigh the measured benefits

Our analysis of the quantifiable benefits associated with CASE projects, combined with the fact that 70 per cent of the PEP

members using CASE claimed that the benefits achieved were less than expected, raises some doubts about the claims made by the suppliers. Unfortunately, most PEP members have no real measures of the benefits that they are achieving through the use of CASE and this limits the validity of the cost/benefit analysis that we can carry out. On the other hand, some members do have an analysis of most of the costs they have incurred.

During an 18-month period from the beginning of 1989, Generale Bank, for example, has installed nearly 50 IEW workbenches from KnowledgeWare, and during the six months from 1991, it has adopted APS from Intersolv. At the time of our research, 90 of its 500 staff had completed their training with APS and 50 with IEW. There are plans to train a further 100 with APS. Generale Bank did not provide the internal costs incurred in assessing and implementing the tools, or the hardware and maintenance costs for APS. However, we have estimated that it will have invested nearly £900,000 in CASE by the end of a two-year period. The detailed costs are shown in Figure 2.4.

A full cost breakdown for a typical US organisation with 200 development staff was given recently in *The American Programmer*. In

Figure 2.4 The costs of CASE are typically several thousand pounds per developer per year

The costs incurred by Generale Bank in Belgium are itemised below.

APS	
Software	£250,000
Training of 90 staff	60,000
IEW	
Software and hardware	120,000
Training of 50 staff	15,000
Annual maintenance	24,000
Training of remaining 100 staff with APS	66,000
Total expenditure on CASE reported by Generale Bank	**£535,000**
Other costs estimated by Butler Cox and suppliers	
Software evaluation and installation costs (APS and IEW)	130,000
Loss of productive time due to training	150,000
APS maintenance costs, consultancy and extra hardware	80,000
Total cost (over two years)	**£895,000**
Average cost per developer per year (240 staff trained)	**£1,865**

The following costs of adopting CASE tools for 200 development staff were itemised in *The American Programmer*.

Initial investment		
Workstations: 200 at $10,000	$2,000,000	
CASE software:		
– Analysis/design tools: 150 at $5,000	750,000	
– Systems planning tools: 10 at $5,000	50,000	
– Management tools: 10 at $5,000	50,000	
– Implementation tools (mainframe)	250,000	
– Maintenance tools (mainframe)	100,000	
Interface/bridge software development:		
– 50 days of consulting at $1,000 per day	50,000	
Professional staff training:		
– Staff time: 10 days each at $200 per day	400,000	
– Trainers: 100 days at $1,000 per day	100,000	
Total investment	**$3,750,000**	
Ongoing costs (per year)		
Hardware maintenance: 200 workstations at $500	100,000	
Software maintenance/upgrades:		
– 10 per cent per year for PC-based software	85,000	
– 15 per cent per year for mainframe software	52,500	
Software engineering group		
– Four people at a cost of $50,000 per year each	200,000	
Ongoing staff training:		
– Two days per person per year at $200 per day	80,000	
Attendance at CASE conferences		
– Three people at four conferences at $1,500	18,000	
Miscellaneous	10,000	
Annual ongoing costs	**$545,500**	
Total cost over five years	**$6,477,500**	
Average cost per developer per year		**$6,500**

the example, the total costs of nearly $6.5 million (£4 million) over a five-year period were broken into an initial investment of $3.75 million (just over £2 million) and an annual ongoing cost of $0.55 million (nearly £350,000). Figure 2.4 gives a complete breakdown.

The benefits to the business need to be very large to justify the high costs of CASE

With such high costs – approximately £800,000 per year for the first five years for 200 staff – the productivity saving, or added value to the business, needs to be very large. From our analysis of the use of front-end CASE tools, we see no such savings in the development process, only a slight improvement in quality. Code generators and fourth-generation languages, however, do offer large savings in the development process and their use should be encouraged.

Clearly, the use of front-end CASE tools within any development department requires careful consideration and assessment. Systems development managers need to assess their real need for CASE and the department's ability to assimilate the changes that CASE will require. Many organisations rush too quickly into CASE and omit these steps. As a consequence, they typically fail to gain the full benefits that can be realised from CASE.

Chapter 3

The department must ensure that it is ready before adopting CASE

Systems development managers need to have a clear understanding about the precise needs of the department before they can assess which kinds of CASE tools will be beneficial to them. They also need to understand the demands that will be made on the department as it implements CASE so that they can assess whether the department is ready to take it on and avoid the pitfalls. Only if systems development managers have done this, can they adopt, use and exploit CASE with confidence.

Compare the department's needs with the potential benefits of CASE

Few systems development managers understand how CASE can help them to meet business needs

Most systems development managers know in which areas their department needs to improve to meet the demands of the business – for example, improving the quality of the development process or increasing staff motivation and morale. Very few, however, really understand how CASE might help them satisfy these demands.

Failing to understand the true relationship between the needs of the department and the benefits provided by CASE typically results in missed opportunities. There are complex relationships between the various benefits associated with CASE and the numerous types of CASE tools. An increase in the level of benefit achieved in one area may well incur costs in another. For instance, the front-end design tools that encourage users to participate should, in theory, result in applications that meet the users' needs better (although we have found no data to prove that this actually happens). As a consequence, there will be fewer requests for changes and a greater level of use. Typically, however, more time and effort is required to use this type of CASE tool and to develop the full specification, so the costs for developing the application will be higher than normal as the main-build time typically remains the same.

A real understanding of the benefits of CASE can be achieved only through measurement within the development department, and through small pilot studies on the effect of CASE tools on the development department and on the current level of benefits being achieved. Measurements for assessing the benefits of CASE are discussed in more detail in Chapter 5.

Assess the readiness of the department to cope with CASE

Many organisations fail to realise that they are not ready to adopt CASE. They recognise the potential benefits that can be achieved,

but fail to realise the extent of the changes that will be required. CASE affects nearly all parts of the development department, and many changes therefore need to be put in place to support it – organisational changes to establish a technical-support team, for instance, or staffing changes to ensure that people are trained in the new methods or that programming staff are developed into analyst/programmers. Unless these changes are planned and managed, the full benefits of CASE will not be realised.

Many changes need to be made in the development department to support CASE

Some managers are able to assess the situation and recommend appropriate changes without any guidance from a method. Most feel more confident with a method to ensure that their investigations and recommendations are complete. During our research, we spoke to several organisations that have adopted methods of some kind. For example, both Sun Alliance, an insurance company in the United Kingdom, and Multihouse, a software services company in the Netherlands, have defined a set of tasks aimed at improving the development process before considering the adoption of CASE, if at all. Although these tasks do not derive from a formal assessment of the department's needs for CASE, they are the types of tasks that have to be carried out with the methods discussed below.

The development process should be improved before adopting CASE

In the past, Sun Alliance had made limited use of both Auto-Mate Plus from LBMS and IEW from KnowledgeWare. However, recently, it has focused on improving development productivity and quality through management initiatives. The systems manager in a 500-strong IBM development department said, "The main drive is to get the culture and the people in place, and right, first. Then, we will look at the tools to support the process." To date, Sun Alliance has implemented several initiatives including a measurement programme, a scheme to improve the planning and control of developments, education of development managers, introduction of inspection techniques and introduction of its own project-management method, based on SDM. It is now looking at the tool marketplace and has initiated a study of development tools likely to be available by 1995. The approach adopted by Multihouse with its Multihouse Projekt Aanpak MPA (Multihouse Project Approach) is very similar. It will be completed by 1992.

Formal methods are available to assess the readiness of a department for CASE. These help to identify potential problems and to manage the process of change. Such assessments should ideally be carried out before an organisation adopts CASE. They do, however, also have a valuable role to play in identifying potential problem areas once CASE has been implemented.

Methods for assessing readiness for CASE are available

We recognise that any method has strengths and weaknesses and that none is a substitute for good management. Methods simply support the management process and help to ensure completeness of thought. While many PEP members recognise the advantages of proceeding systematically, we were not able, during our research, to identify any who have successfully adopted the methods described below. We believe that both of them help in assessing the readiness of the organisation for CASE.

Of the methods that we investigated, two were based on the practical experience of CASE users, typically in the United States, or were in use within several organisations. The first is the use of

Critical success factors help to identify potential areas of risk

critical success factors to direct the attention of management to key areas. They help to identify potential areas of risk and raise the level of awareness of these issues during implementation. Figure 3.1 lists the six critical success factors for the implementation of CASE tools identified by DCE Information Management Consultancy, a Dutch consultancy company.

Figure 3.1 Six factors were identified as critical to the successful implementation of CASE tools

The introduction process must be a formal project and be managed accordingly.
Realistic expectations – requirements for CASE must be fully understood.
Cultural readiness – staff and management must be receptive to new ideas.
Organisational readiness – the right organisational structure must be in place to support the new technology.
Technology readiness – the right technology platforms and approaches must be in place.
The right tools and techniques – there must be a clear understanding of how the existing tool base will affect the use of CASE.
(Source: DCE Information Management Consultancy)

The CASE/IM method reveals areas where change is critical

The second is the CASE/IM method, defined in *Using 'readiness' to guide CASE implementation*, by Howard Rubin. The CASE/IM approach is well developed, has been tried by several organisations in the United States and is easy to complete. CASE/IM has been used with great success to assess organisations' readiness to adopt reverse-engineering tools, maintenance tools and the Information Engineering family of tools. The organisations assessed ranged in size from 8 to 200 systems staff. In all situations, the method revealed areas where change was critical and helped to manage the process of change.

CASE/IM consists of eight pairs of measures, eight for assessing the organisation, and eight for assessing the CASE tool. The features measured are listed overleaf, in Figure 3.2. The measures are represented in a Kiviat chart, which is a means of displaying several related features on the same diagram. Each feature is represented by a radius of a circle. The score for a particular feature is marked as a point on that radius – the lowest score at the centre and the highest at the circumference. The marks on the radii are then connected to form a polygon or footprint (see Figure 3.3, on page 19). When the two sets of eight measures illustrated in the top two Kiviat charts in the figure are overlaid, the gaps are clearly revealed (see lower chart), indicating where attention needs to be concentrated.

The approach used in CASE/IM consists of eight steps. The first three, which are based on the eight pairs of measures, form the assessment, and the other five are directed at the management of change, or the implementation phase of CASE. The assessment steps are as follows:

- Assessment of organisational readiness. This step yields a picture of the organisation's ability to assimilate CASE tools by assessing the eight organisational-readiness features.

Figure 3.2 CASE/IM consists of eight pairs of measures

Organisational-readiness features		Tool-attribute features
Motivation: Level of commitment to improving productivity and quality	<=>	**Benefits**: Level of gain expected
Investment: Willingness to invest capital needed	<=>	**Cost**: True cost to acquire and implement
Skills: Ability to incorporate concepts into work actions	<=>	**Skills**: Skills needed to use tool effectively
Concept knowledge: Knowledge of concepts that are the foundation for using tools and techniques	<=>	**Concepts automated**: Knowledge of development concepts needed to use tool effectively
Culture: Willingness to use new tools	<=>	**Impact**: Breadth of the impact of change – task, process and so on
Organisation support structure: Appropriateness of the support structure for introducing new tools	<=>	**Support needed**: Mechanism needed to ensure effective use and penetration
Technology: Technology in place today	<=>	**Technology**: Technology that must be in place
Applicability: Dominant work focus (for example, new development, support)	<=>	**Work spectrum**: Focus on development/maintenance

- Assessment of organisational context. In this step, the organisation's expectations and capabilities are mapped against what is required for the CASE tool(s) to be adopted successfully. The CASE tools are assessed with the eight tool-attribute features.

- Gap analysis. This step indicates the extent of the gap between the organisation's readiness and the prerequisites for successful adoption of CASE tools.

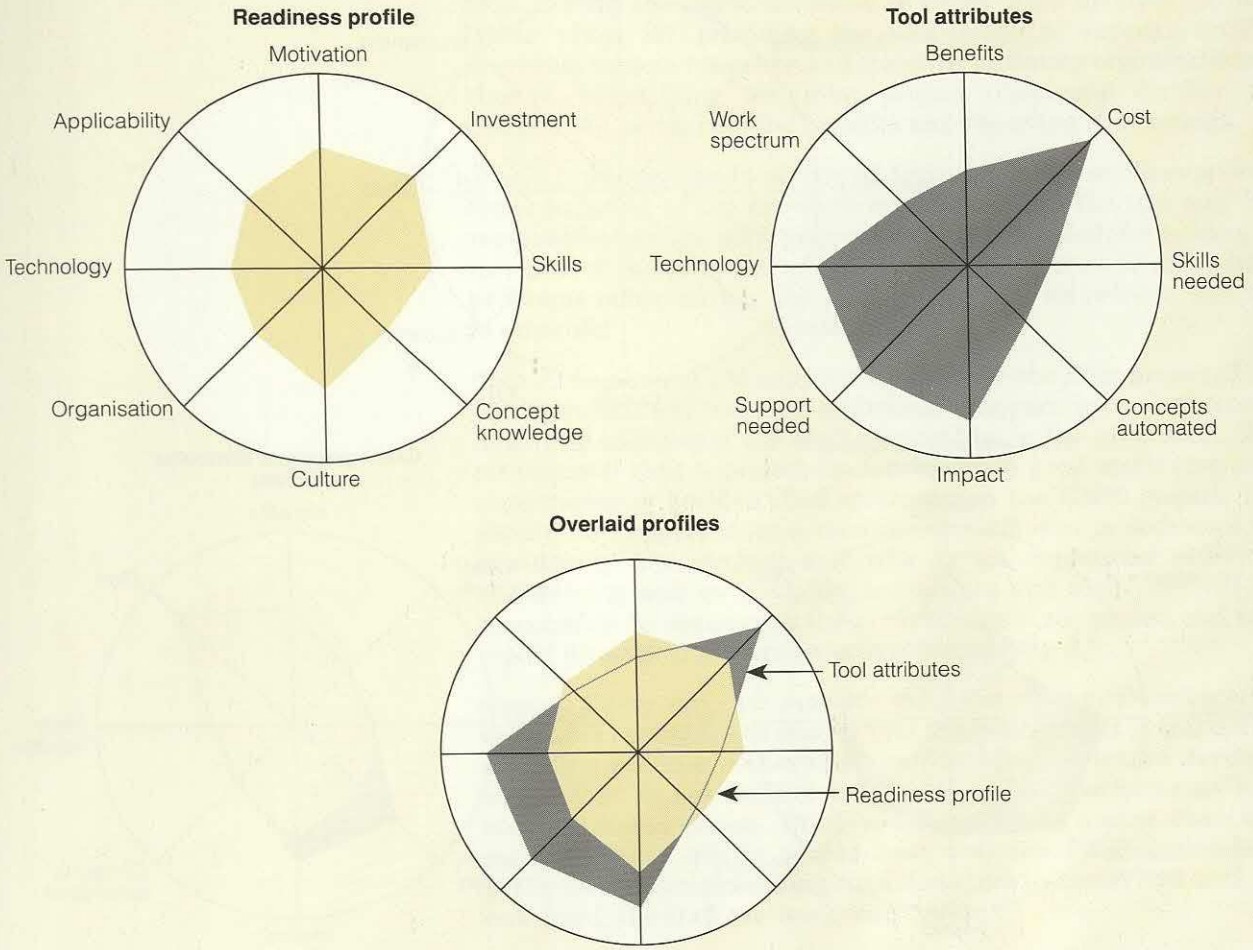
The basic questions used to assess the organisation and the tools are reproduced in Appendix B. The remaining five steps cover implementation risk analysis, planning, support, tracking and evaluation. The process of implementing CASE is discussed later in this chapter.

Rubin completed a survey of 21 organisations in the United States and produced an average profile for the group. This is shown at the top of Figure 3.4, on page 20. The four Kiviat charts in the lower half of the figure show how well this average profile matches the profiles of four different types of development tool – a front-end tool, a code generator, a fourth-generation language, and an integrated CASE tool. The grey areas show where there is a gap between the requirements for the tool and the capabilities of the department. These grey areas are the ones that require investigation and action before adopting, or care while using, the development tool.

The results of Rubin's survey confirm the results of our analysis of the PEP database. Rubin's analysis shows that the average

Figure 3.3 Kiviat charts show those aspects of the organisation and of the tools that have been assessed

In the diagram showing the overlaid profiles, the grey area visible on the NE axis indicates that the cost of the tool exceeds the organisation's willingness to invest. The grey areas visible on the W, SW and S axes highlight that there are some problems with the development technology in place, with organisational support for the tool and with the organisation's willingness to use the tool. The absence of grey areas on the other axes indicates that the company is 'ready' for CASE in all other respects.



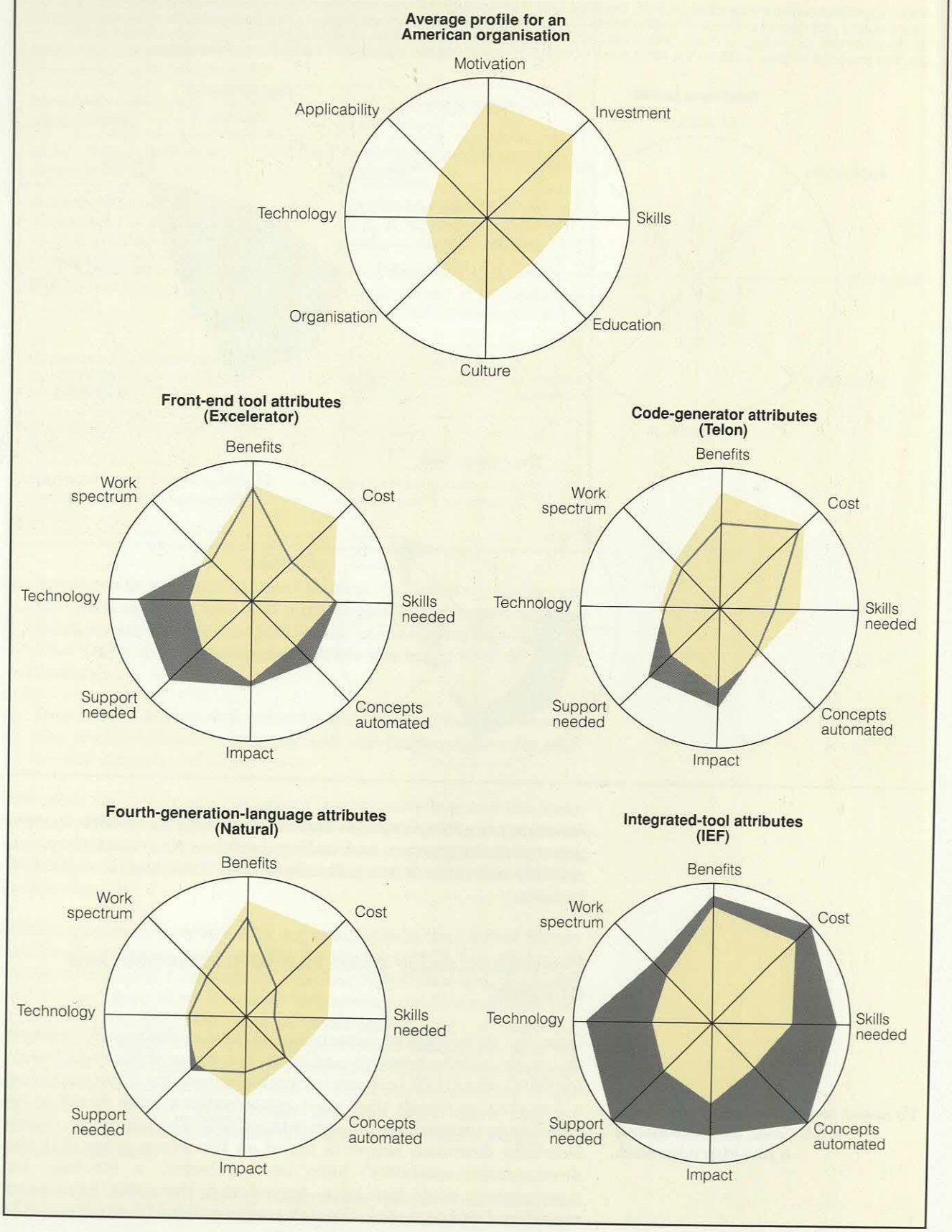
American organisation can exploit and assimilate both fourth-generation languages and code generators, but that they are unlikely to be able to assimilate front-end CASE tools or full CASE toolsets.

Systematically plan the implementation of CASE

To avoid implementation problems with CASE, we recommend a five-step approach

Once the development department is sure of the match between the needs and the benefits, and of the readiness of the department for CASE, the CASE tool can be implemented. We have identified a five-step approach that most organisations have found to be effective in successfully implementing CASE. This approach, which is briefly described below, is based on the assumption that the development method(s) have been selected, a strategy for development tools has been agreed and the risks have been considered and assessed. The risk analysis should include possible

Figure 3.4 The average US organisation is not ready for an integrated toolset



organisational, political, personal, technological and business barriers to implementation. As with most new products, there will be some resentment from staff who are more oriented towards the traditional approach to development. It is therefore critical that CASE is implemented without any major problems.

Step 1: Plan the implementation of CASE. This involves formulating plans for providing the technology to support CASE, providing human resources and training, initiating organisational change, identifying the pilot project (discussed further in Chapter 4), measuring the benefits and reporting the findings.

Step 2: Implement and test the CASE tool received from the supplier. Some tailoring of the development method and the tool may be required before the pilot project is started. It is helpful to have a member of the pilot team involved in this process. If and when problems occur during the pilot project, his knowledge will be invaluable.

Step 3: Implement the support required for the pilot project. The pilot project will require additional support in many areas, including training of the staff who will be in the pilot team. We recommend that a project leader who is a good motivator and communicator be identified to champion the CASE project. He should be supported by a member of staff who is dedicated to monitoring the project and who is an expert on software engineering and on methods, techniques and tools. He will be responsible for measuring the performance of the project and will report directly to the review group during Step 5.

Step 4: Carry out and monitor the pilot project. Development managers should try to ensure that the pilot project is treated as a routine exercise. Otherwise, staff involved in future developments that do not achieve the same levels of performance will become disheartened. Staff must be made aware that the measurements are to assess how well the CASE technology performs in normal working conditions, not to assess how well, or how hard, the staff can work with CASE.

Step 5: Review or evaluate the whole process. This review should assess both how well the pilot project was carried out and how well the method and CASE tools performed. These combined assessments will show how successful the adoption of CASE has been. In the light of this, plans for the continued support of CASE should be amended and updated.

Many development departments have now adopted CASE. Several have managed to gain some benefits; many more have not. In the next chapter, we look at the initiatives that have been taken to support the successful implementation and use of CASE.

Chapter 4

To achieve the benefits, CASE must be fully supported

There have probably been more reports of the failures associated with CASE than of the benefits. Most of these failures are not directly associated with the CASE technology itself; they are a failure of the initiatives supporting CASE. In her book, *CASE is Software Information*, Carma McClure identifies 11 main causes for the failure of CASE. They are listed in Figure 4.1. As CASE is a benefit multiplier, as described in Chapter 1, the initiatives put in place to support the technology are as important as the technology itself. If the approach recommended earlier in this paper is adopted, many of these initiatives will be in place before CASE is implemented.

Most failures with CASE derive from a lack of supporting initiatives

Figure 4.1 Eleven main causes have been identified for the failure of CASE

No method or standards in place
Ignoring the importance of management
Too much emphasis on the CASE tool as the 'silver bullet' solution
Confusion about what the CASE tool does
Misuse of the CASE tool
Perception of CASE as a risk
Unwillingness to change current methods
Uncertainty (or lack of consensus) about the problem that CASE is trying to solve
Poor integration of tools
Inadequate functionality
Poor documentation and/or training

(Source: McClure, C. *CASE is software information.* London: Prentice-Hall International, 1989)

The initiatives described in this chapter show what successful users of CASE have done to support the technology in order to gain the benefits. Obviously, these initiatives will vary in complexity, variety and cost from one organisation to the next and from one CASE product to the next. None is new or revolutionary. In some areas, the initiatives quoted by different organisations as central to their success are contradictory. PEP members should assess the appropriateness of particular initiatives for their own specific situation.

In this chapter, we make frequent references to the organisations consulted in our research. Figure 4.2 summarises the different development environments within the European organisations so that they may be easily compared.

Figure 4.2 Organisations quoted in this chapter use a wide range of CASE tools

Organisation	Business sector	Country	Development staff	Hardware	CASE software*
BP Oil	Refining and marketing	UK	100	Unix	PREDICT-CASE
CCHA Housing	Service	UK	14	IBM	Synon/2, IEW, POSE
Eastern Electricity	Utility	UK	100	Bull	Auto-Mate Plus, PACBASE
Electrabel	Utility	Belgium	100	IBM	IEF
Extel Finance	Service	UK	180	IBM Other	IEW, Bachman DB2 design toolset
Generale Bank	Banking	Belgium	500	IBM Digital	IEW, APS
Norwich Union	Insurance	UK	700	IBM	FOUNDATION, IEW, Micro Focus Cobol Workbench
Prudential	Insurance	UK	600	IBM	Auto-Mate Plus, PREDICT-CASE
Rowntree Mackintosh	Manufacturing	UK	90	IBM Digital	ORACLE
Volkswagen Audi	Distribution	UK	50	IBM	IEF, APS
Yorkshire police	Service	UK	10	Data General Bull	ORACLE
Yorkshire Water	Utility	UK	80	Digital ICL	IEW, CorVision

***CASE software**
PREDICT-CASE
Synon/2
IEW
POSE
Auto-Mate Plus
PACBASE
IEF
Bachman DB2 design toolset
APS
Telon
FOUNDATION
Micro Focus Cobol Workbench
ORACLE
CorVision

Suppliers
Software AG
Synon
KnowledgeWare
Computer Systems Advisers
LBMS
CGI Systems
JMA Information Engineering
Bachman Information Systems
Intersolv
Computer Associates
Andersen Consulting
Micro Focus
Oracle
Cortex

Do not underestimate the cost and importance of education

Extensive education and training will be required

Not only are the costs high, but the scope of the education and training that is required is extensive and it will be time-consuming. It will also require careful planning.

The costs and the breadth of education and training required for CASE are typically underestimated. The Maryland Casualty Company, an insurance company in the United States, found that during the first year, training, data administration seminars and consultancy services represented 37 per cent of the total costs for CASE. In other organisations, education alone has accounted for up to 50 per cent of the cost of CASE.

The education and training required for CASE is not just about the technology. Nor is it simply a matter of sending staff on courses. It should also aim to manage expectations and to win the commitment of both managers and users. Generale Bank, for example, issues a regular newsletter to keep everyone in the organisation informed and up to date with CASE. It believes that it is essential to keep everyone committed to it because of the scale of investment that it will have to continue to make to get the most out of the tools.

It is essential to manage the expectations and win the commitment of managers and users

There is a long learning curve for everyone involved. Rowntree Mackintosh, a UK-based confectionery manufacturer and distributor, aims to spend 10 days per member of staff on training. This comprises five days for the method, one day for the front-end CASE tool, two days for general office facilities, and two days for the use of the dictionary. This represents a potential investment of 700 working days, plus any course fees, for 70 development staff. At Volkswagen Audi (UK), which is responsible for the distribution of all Volkswagen, Audi and MAN vehicles and parts in the United Kingdom, developers are given 10 days' training, and it is typically three to six months before they are competent with front-end CASE.

Education and training for CASE should be planned and managed from the outset. When developing plans for CASE, systems managers should define the types of education and training to be provided for particular groups of people. The education provided by suppliers is typically sufficient, but cannot easily be tailored to meet the needs of a specific organisation. In addition, if different tools are being adopted for different life-cycle stages, it is very difficult to find one supplier able to provide adequate training across the whole life cycle. In some cases, therefore, special provisions will need to be made. For example, several organisations stress that staff who use front-end tools require strong analysis and design skills as well as knowledge of the technical aspects of the database and the languages being used. Some special training may need to be offered to such staff. Others have found that trainees often respond better if they work in project teams that include experienced staff. Such considerations should be borne in mind when training plans are being devised, as should the timescales, the costs, and the option of providing at least some of the training in-house.

Education provided by suppliers is not organisation-specific

Choose a high-profile project first

For many, the choice of application for the initial use of a new technology is predetermined – the timing of the introduction of the technology means that there is no choice. Where there is a choice, it is sensible to select a project that is likely to be able to be completed to plan, and that has a high profile in both the development and the business environments. There is little to be gained from successfully completing an initial project that has limited business impact.

The first project should have a high business impact

Two of the organisations we investigated during the research – Eastern Electricity, an electricity-distribution utility in the United Kingdom, and Extel Finance, a UK-based financial information service – both gained rapid acceptance of CASE by using high-profile or strategic projects for the initial project. This is obviously

a high-risk approach, but if it works, it results in widespread knowledge of the success and an increased rate of acceptance of CASE. There is nothing like business necessity for ensuring that developers get the most out of CASE. The systems development service manager at Extel Finance said, "If it had not worked, I would not be in this job today".

Recognise the value of an effective user

Users should be encouraged to participate in systems-design decisions

Although many CASE tools require technically skilled operators, the users' role with front-end CASE tools is very important. Recognising this can prove to be very beneficial. Yorkshire Water, for example, when discussing the designs for a system, displays the screen images created by the CASE tool on a large projection screen or wall. This increases the number of staff who are able to view and discuss the information and makes joint decision-making, by the team and users, much easier. This encourages users to be more interactive, and keen to be involved with the project.

This greater level of involvement with users is important but does not happen automatically. Yorkshire police realised that CASE was not only a shock for the developer but also for end users. They recognised that there would be a certain amount of apprehension, and possibly unwillingness, among users to commit to an approach that they did not understand. To overcome this, the development department maintains a continual dialogue with users, who are part of the development team.

Choose the right development method

It is essential to implement the development method before adopting CASE tools

It is widely recognised throughout the industry that an appropriate development method should be adopted before the learning process is initiated with CASE tools. Some organisations, however, still hesitate because the implementation of a method can take two to three times longer than learning to use the tool and seems to delay the whole process. Whatever the disadvantages, however, a method is essential for the success of CASE. CCHA Housing went so far as to say, "Once a sound method is in place, the department should stick to it, even if it means that several of the staff, typically the sceptics, leave".

Have a single, central dictionary

Lack of integration between dictionaries can cause great difficulties

The dictionary, or repository, is at the heart of the CASE environment, and a single, central dictionary is the best option. Several organisations have adopted more than one dictionary and most have found the lack of integration between them very limiting. At the Prudential, a UK-based insurance company, information is being passed manually from one tool to the next – the company plans to create a central dictionary as its next step. BP Oil, the UK-based oil refining and marketing company, stopped using IEW in favour of Software AG's PREDICT-CASE. The deputy-director of BP Oil's European Systems Programme stated, "PREDICT-CASE gives us things our previous CASE tool did not in terms of a centralised repository". Norwich Union also

recognises the fundamental importance of the integrated dictionary, but is experiencing difficulties with the interfaces to it.

Although almost all the organisations we spoke to agreed on the importance of the dictionary, there was some disagreement on its use. Some, such as Electrabel, a utility company in Belgium, stressed the importance of completing a business analysis model as the key to its success. Some, such as CCHA Housing, felt that the nine months it spent developing a business model at the outset was too long for an organisation of its size. Others are populating the dictionary in a staged manner, depending on the areas in which they are doing work. Most organisations agreed that a database administrator should be appointed at the start to avoid having to retrofit dictionaries and having to resolve duplication issues.

A database administrator should be appointed to control the dictionaries

Control the use of CASE tools to ensure their effective use

Systems development managers must monitor how CASE is used within the development department – assessing whether it is appropriate to use CASE tools at all, and if so, to what extent. This assessment will vary depending on the type of CASE tool and the application being developed. Getting the balance right will ensure that maximum benefits are achieved.

Control of the use of the tool is considered very important at a major UK supermarket chain. It has found that all code generators have their strengths and weaknesses. The development support manager stated, "If these are not respected, productivity may well drop, and the product [Telon] gets rubbished by the workforce". One team designed screens that were so complex that all development productivity gains were completely lost. This area is still seen as one of the most common problems today.

With uncontrolled use of CASE tools, productivity gains may not be realised

Extel Finance warns of the need to control the front-end design process carefully, because the CASE tool tends to encourage endless iterations of the design. The systems development service manager warns, "Like most technologies, it seduces technicians. It is not a tool for doing the first pass at data modelling. At first, too much time was being spent at the front end; now, we do the modelling on a white-board until we are pretty close to what we want. Then, we start using IEW." Both Norwich Union and Volkswagen Audi (UK) had problems when the use of CASE tools allowed developers and users to expand the scope of applications unnecessarily.

CASE tools tend to encourage endless iterations of the design

Although it is true that the CASE tools are used to support the chosen method, it is not always true that, once the method is in place, the tools can be tailored to fit. One lesson learned at Eastern Electricity was that it is essential to plan the best way to use the tool rather than bend the tool to fit existing working practices. This process needs careful management if the tool is not to be over-used in a sub-optimal manner, or under-used as a result of the temptation to revert to the old tools (usually Cobol) whenever a problem arises.

The tool must not be tailored to fit the development method

Chapter 5

Measures are essential to manage costs and monitor benefits

The importance of measurement in the development department is regularly acknowledged but measurements and their interpretation remain a neglected area of the development process. As one CASE expert recently stated, "Measurement is the weakest thing we do. If we don't measure, we're telling our senior management, 'trust me'." In this paper, we have shown how some of the PEP measures can help the development manager to assess and judge the impact of CASE. However, for a measurement programme to be successful, it needs to encompass the whole development process, not just CASE.

Measurement programmes need to be carefully thought through and implemented. Below, we look at the specific measurements that can be used to help assess the impact, both in cost and benefits, of CASE. Clearly, these are a sub-set of the measurements used in a complete development measurement programme.

Costs can be broken down into three categories for tracking purposes

Costs must be recorded in detail

In Chapter 1, we detailed the average cost per developer of adopting CASE tools. While such calculations are useful in providing an indication of the level of investment likely to be required, they are of limited value to individual organisations because the actual cost per developer can vary widely. To get a clearer idea of the actual costs being incurred, organisations need to record them in detail. How costs are monitored depends highly on the policies within the organisation and how development departments monitor and record costs. Some will divide costs between external and internal, others between projects or budget areas and so on.

We recommend breaking costs down into three categories, as illustrated in Figure 5.1. This will make it much easier for systems

Figure 5.1 The full costs for CASE can be broken down into three categories

Initial	Capital	Recurring
Evaluation of CASE	CASE software	Software licence fees
Installation of hardware	CASE hardware	Software upgrades
Installation of software	– Workstations	Supplier liaison
Supplier support	– File servers	CASE/methods expertise
Training in the method	– Network	Hardware maintenance
Training in the CASE tools	Support software	Refresher training

managers to present a business justification for the investment and it ensures that managers are aware of the commitment that they are making for the future. The three categories are:

Initial costs are costs that arise during implementation or that are incurred to ensure that development staff have at least a basic understanding of CASE. These may vary significantly from organisation to organisation. The initial cost may, for example, double if extensive training is required in the methods supported by the CASE tools. The basic on-site training for 40 staff with the two CASE tools, Excelerator and APS, from Intersolv, for instance, would be a minimum of £18,000. If training in the development method were also required, the additional costs would be a minimum of £60,000.

Capital costs are those that can usually be written off over several years. They typically make up 50 per cent of the costs incurred during the first year and include all the expenditure on workstations, networking equipment and software for both CASE and the supporting environment, such as Windows software. For a company in the United Kingdom, all expenditure on CASE hardware and software can be written off at 25 per cent per year against its taxable profit. The levels of benefits and accounting policies and procedures do change from year to year, however, and are different in each country but they do apply in the majority of countries where there are PEP members, including Australia, Belgium and the Netherlands.

Recurring costs are the costs associated with the continued support of CASE. These will typically include software licence fees, maintenance contracts, user-group membership, refresher training and education, and so on. These costs are typically much smaller than the initial costs and capital costs during the first year.

Most costs are not once-off; the CASE environment will need to be continually enhanced

These three areas of costs are not once-off costs. For most organisations, the CASE environment will require enhancing or changing to exploit the latest CASE technology or to support an additional development method. For instance, West Bromwich Building Society in the United Kingdom is now investing in Software One's tool to link two CASE products that it has been using – ICL's Data Dictionary System (DDS) and Intersolv's SSADM version of Excelerator. The information systems manager said, "We needed to provide tighter integration of the systems development process by linking the two tools automatically. Information passed manually involved rekeying, which led to delays and inaccuracies." Other organisations will face problems in the future due to the use of several different dictionaries that will need to be linked or integrated to remove duplicated information, definitions and so on. Clearly, this will result in additional costs.

This need for continuing investment is not likely to decrease in the foreseeable future, as CASE evolves to the point where it can support several development methods and tools. Eventually, the term CASE will probably also cover iterative development and tools, prototyping development and tools, and object-oriented development and tools. Indeed, in June 1991, Digital Equipment claimed that COHESION, its CASE product set, can now bridge the two worlds of object-oriented development and CASE. The value of

such enhancements depends on how well the CASE environment covers the new method and tools. These enhancements need to be carefully assessed as many more suppliers will undoubtedly follow in this direction.

Few development managers were able to give a complete breakdown of CASE costs

By breaking them down into initial, capital and recurring, the costs associated with CASE can be better monitored, managed and controlled. Many of the development managers we spoke to during the research were unable to provide a complete breakdown of the costs incurred in adopting CASE. Given that the costs incurred during the first year are typically hundreds of thousands of pounds and that the total investment is likely to be millions of pounds, this is shocking.

Measuring the benefits will indicate where greater effort is needed

Few organisations systematically analyse the measures they collect

Experience has shown that measurement is one of the keys to exploiting CASE systematically. Our research for this paper indicates that most organisations regularly collect various measures on the projects they develop. Very few, however, systematically analyse the measures or collate them into meaningful metrics. (*A measure* is of a fundamental dimension or count, typically expressed in simple units, such as hours or lines of code. A *metric* is the result of combining measures to produce a ratio, such as function points or errors per line of code.)

Unless they take measurements before adopting CASE, many organisations will install CASE tools in situations where they have little chance of gaining any benefits from them. Once they have adopted CASE, they will occasionally stumble over solutions to their problems, but unless they have measures to guide them on where to direct their efforts, a lot of time will be wasted and expectations are bound to be disappointed.

The experience of Volkswagen Audi (UK) demonstrates the value of having measures in place. It currently has a development department of 50 staff, working on predominantly IBM equipment. In the middle of 1988, it acquired two CASE tools, IEF and APS, and started to use them regularly in January 1989. Various measures were used to assess and monitor the development environment. They included the basic measures advocated by PEP as well as several other in-house measures. Volkswagen Audi was able to demonstrate, quantifiably, that over a three-year period, it improved its ability to deliver function to the business by over 60 per cent, improved its productivity considerably in relation to the industry average, and reduced post-implementation maintenance effort by 50 per cent.

Specific metrics are appropriate for measuring the impact of CASE

As the needs for CASE tools, and indeed, the CASE tools themselves, will vary from organisation to organisation, so, too, will the measures and metrics used to assess the impact of CASE. Although the PEP measures and metrics give some useful insights into the impact of CASE, they do not, by themselves, identify the full

benefits that may be sought from the CASE environment. The measures we recommend to assess the impact of CASE are collected at the project level and typically analysed to produce metrics at the department level. These metrics are used to assess where there has been change or improvement. Any noticeable trend or variation will typically require further investigation before detailed recommendations can be proposed.

Metrics should be used to assess the impact of CASE tools

Ideally, the measures should be designed to assess the impact of CASE on the performance of both the development department and the business. In practice, however, trying to assess the business impact of CASE gives rise to two problems. First, it is difficult to collect and compare business-performance measures from different business areas as these measures typically relate to different items, such as sales revenue, orders processed, items produced and so on. Second, it is impossible to relate changes in the business measures back to changes in the development department.

It is difficult to assess the impact of CASE on business performance . . .

Some organisations have tried to overcome these problems by defining an artificial relationship between the performance of the development department and the performance of the business – sometimes known as 'the systems to business exchange rate' – but this is typically either too complex or too simplistic to be useful. Because of this, we recommend that measurements be taken only of the performance of the development department, and that they be used to assess the direct impact of CASE. Development departments should, however, also be aware of the business-performance measures and use these as a *guide* to the effect that improvements in the development department are having on the business. The word *guide* is used deliberately because there are many other factors, outside the development department, that also affect the business-performance metrics.

. . . but the direct impact of CASE on the performance of the development department can be measured

The development metrics form what is sometimes called the *management dashboard* for the development environment, with the metrics forming the dials. Below, we have concentrated on the part of the dashboard that can show the impact of CASE on the development environment. Other metrics may be added, but care must be taken to prevent an easy-to-use 'car-sized' dashboard with a few dials expanding to an 'aircraft-sized' one with hundreds of dials that is too complex for most to interpret.

Since we are proposing methods of measuring the benefits of CASE rather than how efficiently the tools are being used, function points are used as the measure of size. The recommended metrics are:

We recommend eight metrics for measuring the impact of CASE

- *Output rate*. The output of the development department, measured in function points per man-month of effort or per pound spent at both the project and department level. User effort should always be included as the level of user involvement increases with the use of CASE. This will be difficult for many organisations to measure because users typically do not record their time. At the department level, this metric is typically accumulated over a quarterly period to remove the variation introduced by different projects. This metric should be calculated for projects that deliver new functionality to the business. Maintenance projects that do not deliver new functionality, such as conversions and projects

aimed at improving operational performance, should not be included; they are included in a later metric.

- *Speed of delivery*. The rate at which the development department produces function, measured in function points per calendar month at both the project and department levels. At the department level, this is typically accumulated over a quarterly period. This metric applies only to projects that deliver function to the business.

- *Maintenance rate*. The amount of effort required to maintain the applications, measured in function points maintained per man-month. This can be reported at the application-area level, such as finance, sales and marketing, distribution and so on, and by type of change – corrective, perfective or adaptive.

- *Maintenance delay*. The sum of the lengths of time that requests for all changes have been outstanding, divided by the number of requests. This may be reported at the application-area level and by type of change.

- *Maintenance work ratio*. The estimated man-days of effort required to meet all the changes requested during a period (typically one calendar month), divided by the effort spent meeting change requests during the same period. Again, this may be reported at application-area level and by type of change.

- *Error rates*. The number of errors found during the different phases of testing (module, suite, integration), divided by the size of the system in function points. Typically, this is measured at the project level.

- *Reliability*. Calculated by dividing 750 (the approximate number of hours in a month) by the number of failures in the first month.

- *User satisfaction*. The satisfaction of the users with the applications provided, measured on various aspects, such as ease of use. The metrics are calculated by dividing an importance rating by a satisfaction rating.

These metrics should be used to assess the impact that CASE is having on the productivity and quality of the development process, the productivity and quality of the maintenance process, and the match of the system with users' needs.

The impact of CASE on productivity can be assessed . . .

The impact of CASE on the development process should be assessed, in terms of productivity, through the first two metrics – 'output rate' and 'speed of delivery'. An increased rate of output indicates that more function is being delivered to the business by the department for a given cost. Increased speed of delivery indicates that the department is delivering function to the business faster. These two metrics should be interpreted together as they are strongly related; an increase in the speed of delivery may result in a decrease in the rate of output.

. . . as can the impact on quality

In terms of quality, the impact of CASE on the development process should be assessed through the two metrics 'error rates' and 'reliability'. These two metrics together indicate the relationship between the effectiveness of the testing process and the number of

errors in the system. These two metrics should be interpreted in each possible combination:

- *Both poor:* Indicates a poor-quality development process (excluding testing), inadequate testing procedures and a poor-quality delivered system.
- *Poor reliability only:* Indicates a potentially poor-quality development process (excluding testing), inadequate testing procedures and a poor-quality delivered system.
- *Poor error rates only:* Indicates a poor-quality development process (excluding testing), adequate testing procedures and a good-quality delivered system.
- *Both good:* Indicates a good-quality development process (excluding testing), adequate testing procedures and a good-quality system.

The impact of CASE on the maintenance process can also be measured

The impact of CASE on the maintenance process should be assessed, in terms of productivity, through the three metrics 'maintenance rate', 'maintenance delay' and 'maintenance work ratio'. Increases in the maintenance rate indicate that a greater amount of maintenance is being carried out for a given cost. Decreases in maintenance delay indicate that the delay between a request for maintenance and its being dealt with is decreasing. A reduction in the maintenance work ratio indicates that the backlog of effort required to satisfy the maintenance requests is decreasing.

In terms of quality, the impact of CASE on the maintenance process should be assessed through the same two metrics used to assess the impact on the development process – 'error rates' and 'reliability'.

It is also possible to measure the impact of CASE on matching systems to users' needs

The impact of CASE on matching the system to the users' needs should be assessed in terms of the two metrics 'user satisfaction' and 'maintenance work ratio'. The higher the satisfaction of users, the better the match between the delivered system and the users' needs. If user satisfaction is low, there will typically be a series of change requests for corrective maintenance, and the total cost of the system will increase.

Using these metrics will enable development managers to assess their use of CASE and its impact. As a result of the interpretation of the metrics, the most appropriate initiatives can be selected and CASE can be exploited and used when it is most appropriate to do so.

Chapter 6

CASE should be adopted with great care

In this paper, we have shown that, although CASE tools have been beneficial to some organisations, they have been a costly failure for many others. In Figure 6.1, we list the actions that systems development managers should take to ensure that implementing CASE is not a disappointing and expensive experiment.

Figure 6.1 Action checklist

Review the costs and benefits associated with CASE tools in your environment:

- Assess the impact of CASE tools on the whole development process.
- Consider the potential impact of CASE over the next few years.
- Compare the benefits and costs and assess the match to the department's needs.
- Consider adopting code generators as a first step towards CASE.

If the review clearly favours CASE:

- Ensure that adequate training is provided to all those using CASE.
- Ensure that the CASE tools support the development method.
- If piloting, consider a high-profile project to stimulate rapid adoption.
- Review the use of all existing CASE tools and ensure that they are used only when they will be effective.
- Avoid using more than one dictionary.

If there is any doubt about CASE, do not adopt it.

The problems encountered in introducing CASE tools are not insoluble

The problems that are commonly encountered by organisations introducing CASE tools are not insurmountable, and for those that are properly prepared, they can provide an environment in which the promised productivity and quality benefits can be realised. Organisations do, however, need to be aware of the impact that CASE will have in order to exploit the technology to the full.

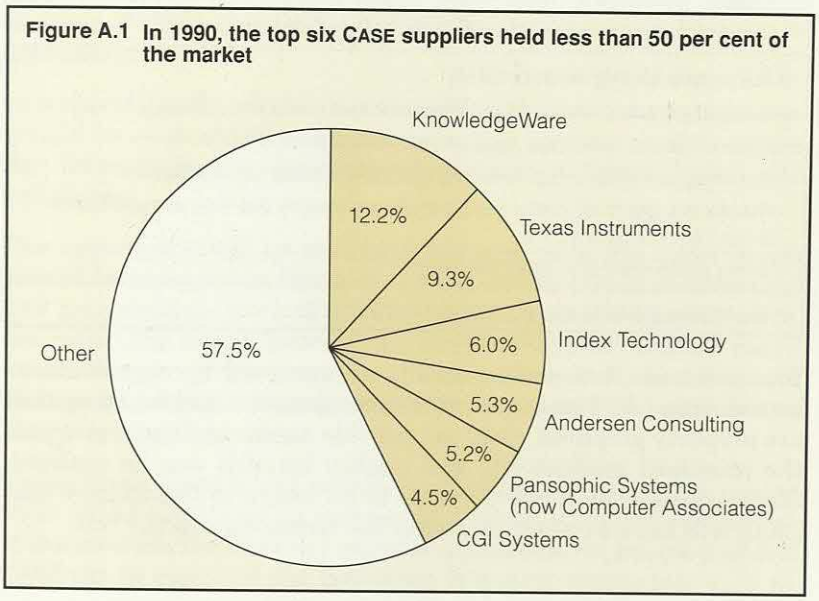
Appendix A

Review of the CASE marketplace

According to various CASE experts, the CASE marketplace is starting to go through a period of rapid change, with an increasing number of mergers and takeovers. This means that many smaller suppliers will either lose market share and collapse, or be taken over. A recent example is the Index/Sage merger, which brings together two products, Excelerator and APS. The new company, which has over 10,000 customers worldwide, is called Intersolv.

In 1990, a report from Strategic Focus on the $750-million worldwide CASE marketplace showed that the top six suppliers held less than 50 per cent of the market. The breakdown is shown in Figure A.1. (It does not feature Intersolv because the Index/Sage merger took place too late for the figures to be included.)

Figure A.1 In 1990, the top six CASE suppliers held less than 50 per cent of the market



As part of the research for this report, we surveyed the suppliers of CASE tools most commonly used by PEP members. Figure A.2 is a summary of the information we received from them.

The market research company, Gartner, predicts that the following suppliers of CASE tools will survive the market shake-out:

- Andersen Consulting with FOUNDATION.
- JMA Information Engineering with Information Engineering Facility (IEF).

Figure A.2 The suppliers of CASE tools are many and varied

	Softlab/Maestro	Computer Associates/ Telon	Oracle/CASE*	LBMS/Strategic Planner, Systems Engineer	JMA Information Engineering/IEF	IPSYS Software/ IPSYS toolset	Intersolv/Excelerator, APS	Information Architechs/ Architech	Ernst & Young/ IEW, ADW	CGI Systems/PACBASE
Supplier/product name										
Company revenue								?		
Product details										
Initial investment		?							?	
Launch date	1980	1981	1988	1989	1987	1991	1983	1990	1985	1988
Hardware	IBM Digital HP ICL Philips	IBM	IBM Digital HP DG Sun	IBM	IBM Digital Unix Tandem	HP Sun	IBM	IBM ICL	IBM	IBM ICL Unisys Bull
Overall product support										
User group	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes	Yes
Help line	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes
Units sold/training and support										
United Kingdom	✓	✓	✓	✓	✓ ?	✓	✓	✓	✓	✓
Germany	✓	✓	✓	✓	✓ ?	✓	✓	✓	✓	✓
France	–	✓	✓	–	✓ ?	✓	✓	–	✓	✓
Benelux	–	✓	✓	✓	✓ ?	–	✓	–	✓	✓
Rest of Europe	–	✓	✓	–	✓ ?	–	✓	–	✓	✓
Australia	–	✓	✓	✓	✓ ?	–	✓	–	✓	–
United States	–	✓	✓	✓	✓ ?	–	✓	–	✓	✓
Worldwide units sold						✓		✓		

	Company revenue	Initial investment	Units sold	Training/support
–			No units sold	No training/support provided
	>£100 million	>£150,000	>100	
	£10 million - £100 million	£150,000 - £300,000	10 - 100	
	< £10 million	> £300,000	<10	
✓			Units sold, but no figures provided	Training/support provided
?	No information provided	No information provided		No information provided

No replies were received from Andersen Consulting (FOUNDATION) and IBM (AD/Cycle). Brochures only and no replies to the questionnaire were received from Cognos (Powerhouse), Digital (COHESION) and ICL (QuickBuild).

(Source: Questionnaire sent to CASE-tool suppliers)

- KnowledgeWare with Information Engineering Workbench (IEW/ADW).
- Oracle with CASE*.

Russel Jones, a CASE industry expert, believes that the European companies, CGI Systems with PACBASE, and possibly Softlab with Maestro, should be included. In the United Kingdom, LBMS may become a worldwide player. Suppliers who sell CASE-related technology, like databases, will also retain a strong presence in the CASE marketplace. Examples include Software AG and Cincom Systems.

Other suppliers that Russel Jones believes are likely to maintain their places as 'niche' suppliers include both front- and back-end suppliers. Front-end suppliers will be Intersolv with Excelerator, and possibly Computer Systems Advisers with POSE. Back-end suppliers will be Computer Associates with Telon, Cognos with Powerhouse, and Intersolv with APS.

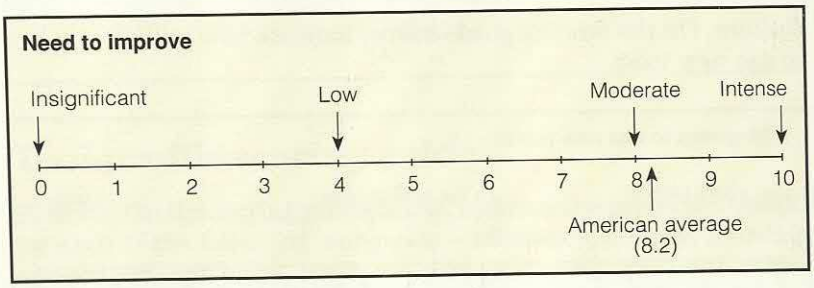
Appendix B

The CASE/IM profile questions

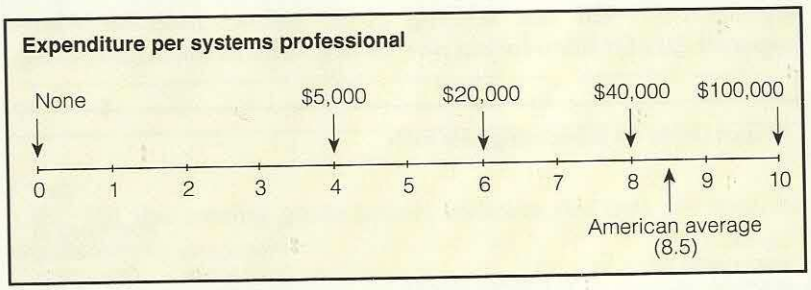
The organisational-readiness profiles and the tool profiles discussed in Chapter 3 are based on answers given to the questions listed in this appendix. Members considering using this approach should contact Howard Rubin Associates in New York for more detailed information.

Scoring guide for organisational readiness

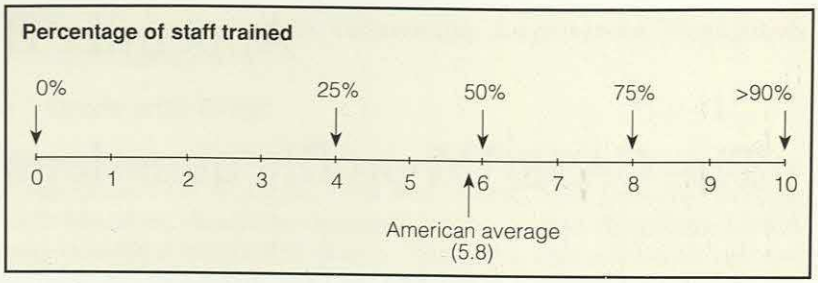
Motivation. On the scoring guide below, indicate the extent to which you need to improve in each of the following categories – efficiency, quality, schedule compliance, flexibility, improved project management and other. Use the average of all the scores as your motivation score.



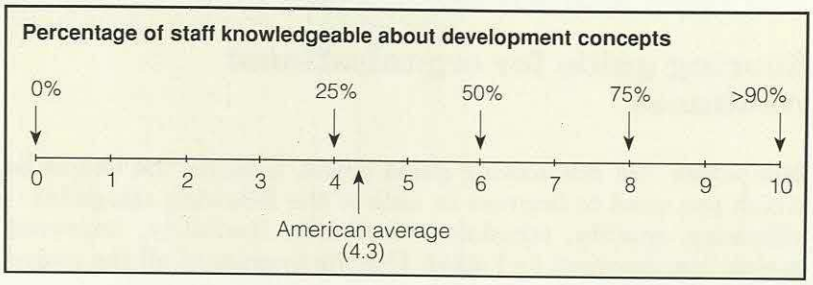
Investment. On the scoring guide below, indicate the level of expenditure per systems professional that you are prepared to make.



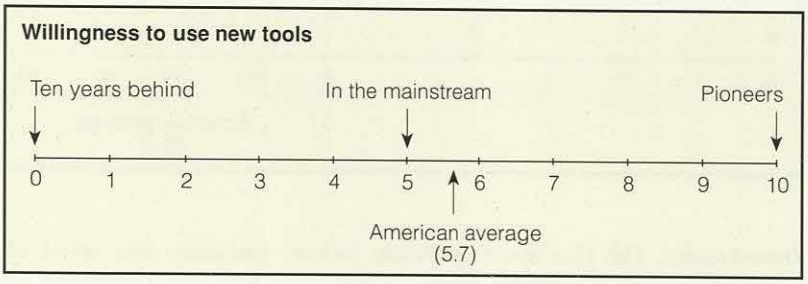
Skills. On the scoring guide overleaf, indicate the percentage of your staff who are trained in each of the following categories – structured analysis and design, structured programming, and the software development life cycle. Use the average of all the scores as your skills score.



Knowledge of concepts. On the scoring guide below, indicate the percentage of your staff who are knowledgeable about each of the following development concepts – data flow, entity relationship, and information engineering. Use the highest score as your score for knowledge of concepts.

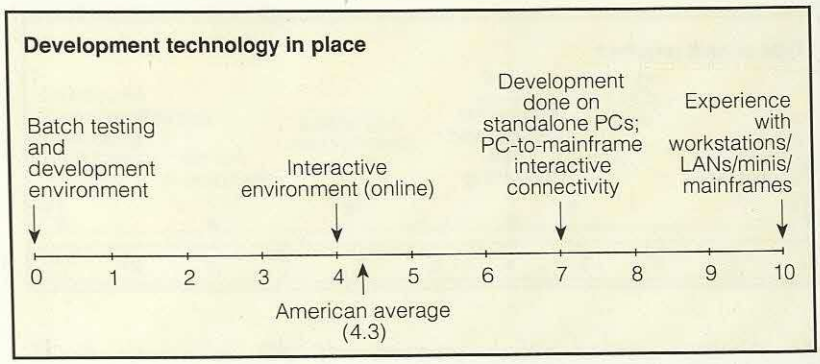


Culture. On the scoring guide below, indicate how willing you are to use new tools.

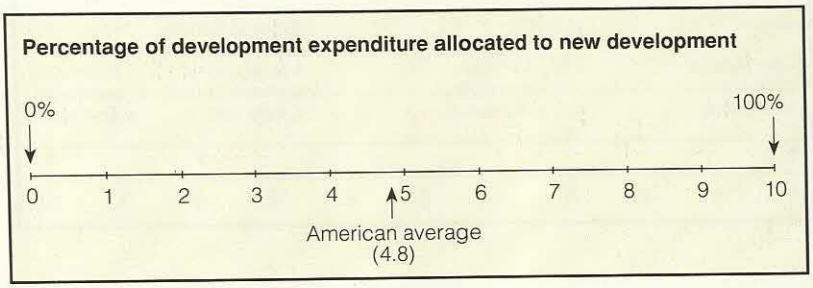


Organisation. On the scoring guide below, indicate where responsibility for introducing new tools resides in the organisation.

Technology. On the scoring guide below, indicate what kind of development technology is in place.

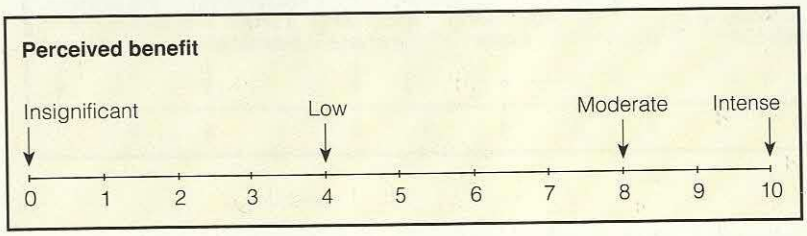


Applicability. On the scoring guide below, indicate the percentage of development expenditure that is allocated to new development, divided by 10.

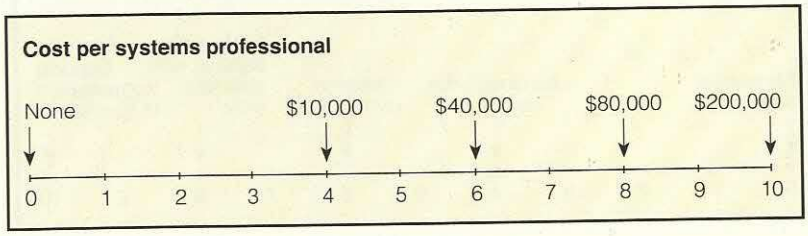


Tool-profile scoring guide

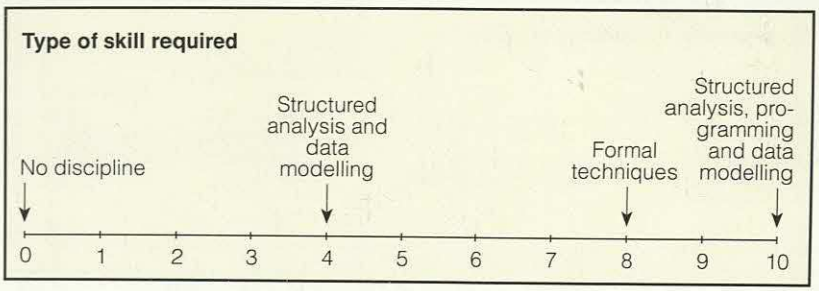
Benefits. On the scoring guide below, indicate the perceived benefit for each of the following categories – efficiency, quality, schedule compliance, flexibility, improved project management and other. Average all the scores to obtain your perceived benefit score.



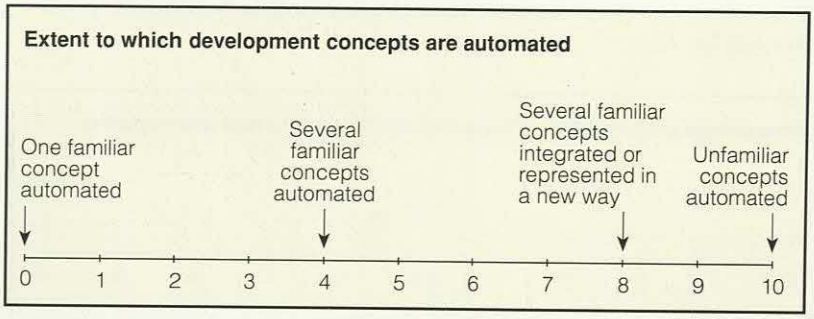
Cost. On the scoring guide below, indicate the cost per systems professional.



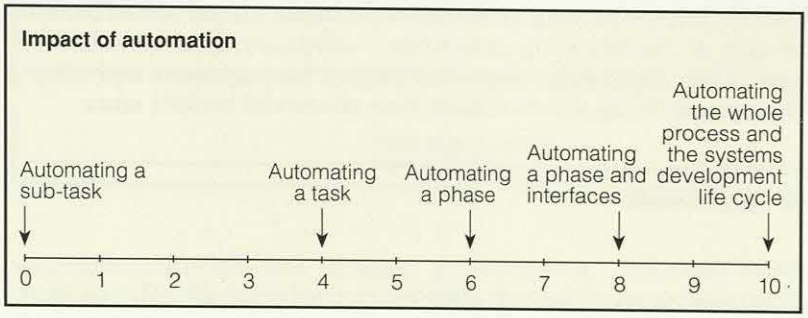
Skills needed. On the scoring guide below, indicate what kind of skills are required.



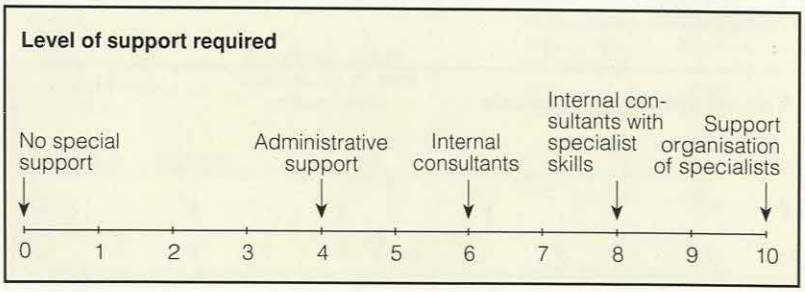
Concepts. On the scoring guide below, indicate the extent to which the tool will automate development concepts.



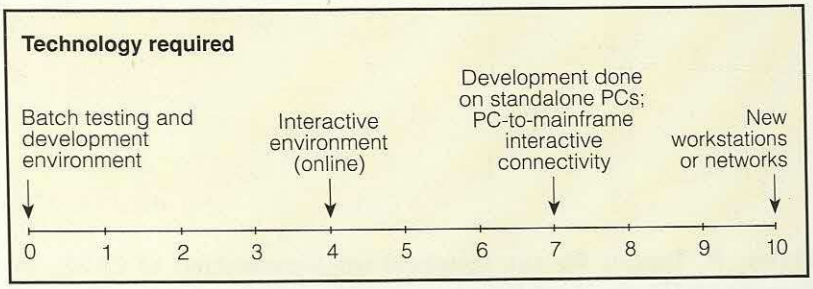
Impact. On the scoring guide below, indicate the impact of automation.



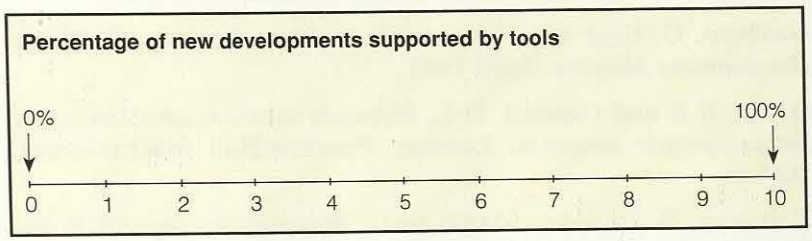
Support needed. On the scoring guide below, indicate the level of support required.



Technology. On the scoring guide below, indicate what kind of technology is required.



Work spectrum. On the scoring guide below, indicate the percentage of new developments that can be supported by this type of tool, divided by 10.



Bibliography

Daley, R. Tactics for the practical implementation of CASE, in *EuroCASE III: the third European CASE conference: proceedings of the conference held in London, April 1991*. Pinner, Middlesex: Blenheim Online, 1991, p.163-180.

Duncan, M. Business measurement and the measurement business. *System Development*, vol. 11, no. 6, June 1991, p.1-6.

Goodwin, C. User experience with CASE technology. *Software Development Monitor*, April 1991.

Grady, R B and Caswell, D L. *Software metrics: establishing a company-wide program*. London: Prentice-Hall International, 1987.

Holloway, S. *Choosing CASE tools*. Woking, Surrey: DCE Information Management Consultancy, 1991.

Hughes, C T and Clark, J D. The stages of CASE usage. *Datamation*, vol. 36, no. 3, 1 February 1990, p.41-44.

Humphrey, *W S. Managing the software process*. Reading, MA: Addison-Wesley, 1989.

Jones, R. CASE technology update, in *EuroCASE III: the third European CASE conference: proceedings of the conference held in London, April 1991*. Pinner, Middlesex: Blenheim Online, 1991, p.203-213.

McClure, C. *CASE is software information*. London: Prentice-Hall International, 1989.

Rubin, H A. Using 'readiness' to guide CASE implementation. *CASE Trends*, January/February 1991, p.37-41.

BUTLER COX

P.E.P

PEP

The Butler Cox Productivity and Quality Enhancement Programme (PEP) is one of the research services provided by CSC Index. It is a participative service whose goal is to improve productivity and quality in application systems development.

It provides practical help to systems development managers and identifies the specific problems that prevent them from using their development resources effectively. At the same time, the programme keeps these managers abreast of the latest thinking and experience of experts and practitioners in the field.

The programme consists of individual guidance for each subscriber in the form of a performance assessment, and also publications and forum meetings common to all subscribers.

Performance Assessment

Each subscribing organisation receives a confidential management assessment of its systems development performance. The assessment is based on a comparison of key development data from selected subscriber projects against a large comprehensive database. It is presented in a detailed report and is followed by a workshop with Butler Cox specialists, during which an improvement plan is developed.

PEP Papers

Four PEP Papers are produced each year. They concentrate on specific aspects of systems development management and offer practical advice based on recent research and experience.

Meetings

Each quarterly PEP forum meeting focuses on the issues highlighted in the previous PEP Paper. The meetings give participants the opportunity to discuss the topic in detail and to exchange views with managers from other member organisations.

Previous PEP Papers

1 Managing User Involvement in Systems Development
2 Computer-Aided Software Engineering (CASE)
3 Planning and Managing Systems Development
4 Requirements Definition: The Key to System Development Productivity
5 Managing Productivity in Systems Development
6 Managing Contemporary System Development Methods
7 Influence on Productivity of Staff Personality and Team Working
8 Managing Software Maintenance
9 Quality Assurance in Systems Development
10 Making Effective Use of Modern Development Tools
11 Organising the Systems Development Department
12 Trends in Systems Development Among PEP Members
13 Software Testing
14 Software Quality Measurement
15 Application Packages
16 Project Estimating
17 Motivating Systems Development Staff
18 Managing Small Projects
19 Involving Users in Systems Development
20 The Benefits of CASE: Myths and Reality

Forthcoming PEP Paper

Distributed Development

CSC Index

CSC Index is an international consulting group specialising in information technology, organisational development and business reengineering. Its services include management consulting, applied research and education.

Other research and advisory services offered by CSC Index in Europe are:

Index Vanguard explores the strategic impact of emerging information technologies.

Quantum, a joint offering of CSC Index and Michael Hammer, researches issues for executives involved in business reengineering.

Directors' Forum provides an exclusive advisory service for the most senior information systems executives of large European organisations.

PRISM, a joint offering of CSC Index and Hammer and Company, provides research on information management issues for the senior information systems management team.

The Butler Cox Foundation offers advice and insight into best practice in information systems management.

Butler Cox plc
Butler Cox House, 12 Bloomsbury Square,
London WC1A 2LL, England
☎ (071) 831 0101, Telex 8813717 BUTCOX G
Fax (071) 831 6250

Belgium and the Netherlands
Butler Cox Benelux bv
Prins Hendriklaan 52
1075 BE Amsterdam, The Netherlands
☎ (020) 6 75 51 11, Fax (020) 6 75 53 31

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

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

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

Finland
TT-Innovation Oy
Sinikalliontie 5, 02630 Espoo, Finland
☎ (90) 358 0502 731, Fax (90) 358 05022 682

Ireland
SD Consulting
8 Clanwilliam Square, Dublin 2, Ireland
☎ (01) 764701, Fax (01) 767945

Italy
RSO SpA
Via Leopardi 1, 20123 Milano, Italy
☎ (02) 720 00 583, Fax (02) 86 45 07 20

Scandinavia
Butler Cox Foundation Scandinavia AB
Jungfrudansen 21, Box 4040, 171 04 Solna, Sweden
☎ (08) 705 83 60, Fax (08) 730 15 67

Spain and Portugal
T Network SA
Núñez Morgado 3-6°b, 28036 Madrid, Spain
☎ (91) 733 9910, Fax (91) 314 3198