An empirical evaluation of the use of CASE tools

An Empirical Evaluation of the Use of CASE Tools

S.C. Stobart[1], A.J. van Reeken[2], G.C. Low[3], J.J.M. Trienekens[4], J.O. Jenkins[5],
J.B. Thompson[1] & D.R. Jeffery[3]

[1]School of Computing, University of Sunderland, England.
[2]Department of Business Administration, University of Limberg, The Netherlands.
[3]School of Information Systems, University of NSW, Australia.
[4]Graduate School of Industrial Engineering and Management Science, Eindhoven University of Technology, The Netherlands.
[5]Department of Business Computing, City University London, England.

Abstract

Investigations conducted in three separate countries by four research groups on the use and effectiveness of CASE within industry are reported. Comparisons are drawn between the three set of research findings and similarities and differences in the impact of CASE within the different countries are identified. The report combines the experience of the authors, and provides a wider insight into the use and effectiveness of CASE tool on a "worldwide" scale.

1: Introduction & Method

This paper reports on research undertaken into the use and effectiveness of CASE tools within the commercial computing business/engineering sector. The results of research activities from three countries: Australia, England and the Netherlands are examined in order to identify common and differing trends of the use of CASE.

In the Netherlands, research was undertaken by interviewing two senior staff members from each of sixteen large organisations [1]. All were senior staff members in an executive function with many years of experience and could be expected to be able to express perceptive insights. The organisations included major software houses, industrial organisations and government departments. The objective of this explorative research was to find priorities for further research in this subject field. In England a survey [2][3] was conducted of 480 organisations to investigate the use, impact and future requirements of CASE within the United Kingdom.

Among the main objectives of the survey were:

- Identify the current level of CASE use amongst Information Technology departments.
- Determine the extent to which the software life-cycle has been automated.
- Determine whether the introduction of CASE has led to improvements in quality or efficiency of development.
- identify problems with current tools and identify requirements for future tools.

A separate invcstigation [4], also in the UK, was carried out into thc factors affecting the succcss of CASE utilisation. In particular this research attempted to build a model of CASE skill maturity.

In Australia a case study approach of questionnaire and interview data collection was used within four Australian financial institutions [5]. These institutions have dynamic systems delivery environments and have used CASE products to assist in their systems delivery to their client base. Of the four organisations surveyed, one employs 35,000 people and has approximately 2000 information systems professionals. Another employs 2000 people and has approximately 180 information systems professionals.

The following sections of this report detail the overall findings of these surveys. The results are categorised into individual sections with both common and differing findings identified.

2: The Major Findings

2.1: The Adoption and Use of CASE

This section describes the extent to which CASE has been adopted within the commercial computing sector.

1066-1387/93 $3.00 © 1993 IEEE

Within the United Kingdom some 18% of survey respondents were currently using CASE [2]. The remaining 82%, who were non-CASE users, sub-classified themselves into:

- those who were currently evaluating CASE (26%),
- those who had evaluated and dismissed it (14%),
- those who are willing to purchase when they identify a suitable tool (5%),
- those uncertain about CASE in general (11%),
- those who had not evaluated CASE but had still rejected it (26%).

The Jenkins and Majumdar survey addressed only known users of CASE tools.

tools	**software houses**	**other organizations**
diversity	- great diversity for modelling (more than two) - great diversity for project control	- small for modelling (one or two) - small for project control
standards	- none for project control - none for modelling	- none for project control - declining importance for modelling
integration over the life- cycle	not integrated	mutually linked (limited integrated)

Figure 1 - Characteristics of the use of tools

Both in Australia and Netherlands the research targeted organisations who were currently using CASE and thus no indication of the overall use of CASE throughout these countries was identified. However, in the Netherlands, Van Reeken and Trienekens found that software houses used several different kinds of tools to support system modelling, which have little or no integration between them. Figure 1, summarises the findings for tool use, diversity and standards within the organisations surveyed in the Netherlands. It was also found that the number of methods and tools used in the phases of the life-cycle varies considerably. In Australia, Low and Jefferys identified the limited availability of CASE tools as Open System products as hampering adoption.

2.2: Why CASE is Used

This section reports the reasons that organisations identified for their use of various CASE products.

The reasons for CASE implementation may seem obvious at first, with organisations wishing simply to benefit in some way through quality, productivity etc, from the automation of part of software life-cycle. In determining why CASE is used within organisations, Van Reeken et al identified that about 70% of all respondents applied CASE tools and methods mainly in order to gain better control of development activities. It was found that managers stressed the importance of project control, with the overall goal being able to deliver a system on time and within budget. Other reasons identified were product modelling and user participation. In Australia, Jeffery et al identified that CASE was used extensively during system analysis and design specification in the belief that this would result in back-end productivity gains. Within the United Kingdom a variety of reasons were stated by respondents. These ranged from those who believed that CASE was the correct way forward but were vague when defining precisely why, to those respondents that have company policies on examining any new development to determine its worth for their particular situation.

2.3: Productivity, Efficiency and Quality

Although it has been widely claimed that benefits in productivity, efficiency and quality will results from the adoption and use of CASE, we examine whether such productivity and/or efficiency gains have been realised by CASE users.

In the Netherlands, respondents identified the following commonly found advantages that could be gained from the use of CASE products:

- increased productivity,
- reduced development and maintenance costs,
- faster delivery,
- improved documentation.

However, these benefits did not appear to be obtainable for any of their current projects. Currently managers were interested only in meeting costs and project deadlines and that the benefits from the use of tools would be viable only in the long term. It was realised that companies would demand improvements in manageability that will compensate them for the effort and costs that the introduction of a CASE tool will entail. Further differences were found between what benefits a CASE tool was required to achieve. The quality and efficiency of the generated product is more important to those organisations that develop in-house. Whilst, developers within software houses see productivity gains as being of key importance.

In the United Kingdom the respondents to the survey indicated that, in practice, they found the measurement of CASE tool quality and efficiency gains very difficult to undertake in practice. Only, 60% of respondents indicated that they monitored staff quality/productivity rates. The software development areas that were identified as having the largest increase in quality were those of: Code Generation, Prototyping and Project Control. The software development phases that were identified as having the largest increase in efficiency were those of: Program design, Code Generation, Prototyping and Maintenance. No respondents indicated that quality or efficiency was reduced with the introduction of CASE tools.

Within Australia, Low and Jeffery identified that organisations spent more time during the system analysis and design phases in order to reap the resultant productivity gains that would occur with the rear-end of the software development life-cycle phases. One of the organisations surveyed did realise productivity gains in code generation. However, in many instances it was found that this gain was often lost due to the inefficiency of the CASE product. This was due to the fact that the tool used large amounts of processing time to re-compile the code between each design amendment, which in turn resulted in wasted time between code debug and test. The other surveyed organisation found that productivity gains could be attributed to their tool but that these to were lost due to problems of applying the tool to the batch processes under development.

In summary, it would seem that the theorised and promised benefits and advantages from using CASE are the same world-wide. However, all of these are difficult to prove categorically and remain a matter of faith with those that use them. Nevertheless, in both the United Kingdom and Australia respondents realised some productivity gains with the code generation phase of development. Also in the United Kingdom, users identified further productivity and quality gains within the back-end of the software development life-cycle, while in Australia these were expected but unproven. Finally, although some productivity gains can be often attributed to tools the immaturity in tool design often means that these gains are lost while performing other activities when using the tools.

2.4: Implementation of CASE

In this section we address the problems associated with the implementation of CASE tools. We also consider the factors which influence organisations not implementing CASE products.

It the United Kingdom, both Stobart, Jenkins and Majumdar independently found that a number of organisations had implemented CASE products only as an investigative experiment. The use of these tools in large development projects was not practical since full commitment to CASE had not yet been achieved. Respondents complained of the lack of management commitment to tool implementation (mainly due to the high costs) and that without this commitment tool implementation and successful use was doomed to failure. A large variety of reasons why CASE tools had not been implemented in organisations was presented. Figure 2, below, illustrates these reasons. Of these, the problems of costs, lack of management backing and current development methods satisfactory were the most frequent. Not unexpectedly organisations were reluctant to commit themselves to a large CASE tool budget. All organisations who have not committed to CASE tools commented that the costs of tools would have to be substantially reduced before they implemented CASE tools. In general CASE tools were recognised as having the potential to provide a

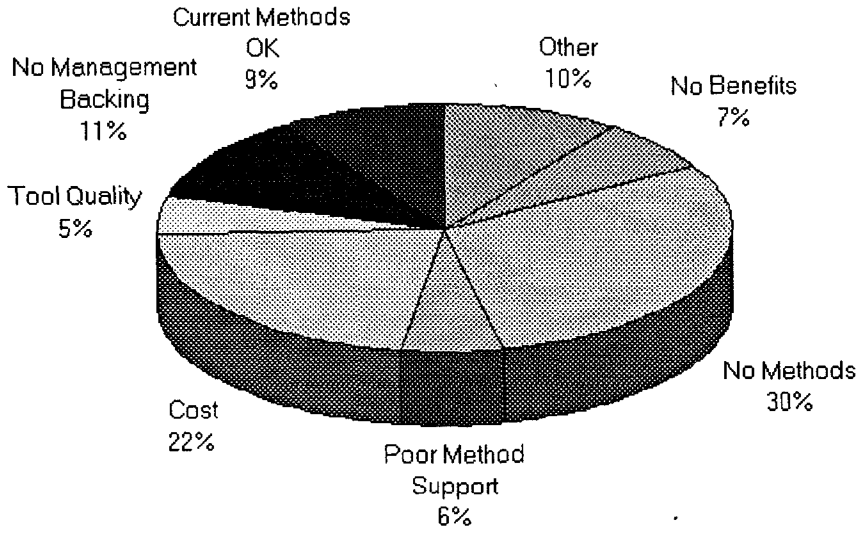

Figure 2 - Reasons for not implementing CASE

major benefit but many were inadequate for the problem which they attempted to address.

In Australia, the surveyed organisations agreed that a well planned training investment must be implemented before CASE can be successfully used. Also, the high cost of what are currently relatively immature tools, has restricted tool use and thus prevented realization of tool benefits. Organisation are similarly unwilling to commit large amounts of money for unproven (if any) returns.

In the Netherlands all companies surveyed were known to be using CASE tools in some form. We found that the costs of tools have not been a problem for these companies. However, implementation of CASE was said to be very dependant upon a well planned training investment.

In summary, it would seem that CASE tools require a well structured implementation plan with both management and developer commitment to the use of tools. High costs of tools seem to be a major reasons for their rejection by many organisations. And finally, without a significant investment in training tool use will be unsuccessful.

2.5: CASE Tools Problems

Current CASE tools vary considerably in their functionality and design. However, many of them suffer from generic problems. This section examines what, these problems are and in what ways they may be stifling the adoption and implementation of CASE tools.

Van Reeken identified that tools have too many inconvenient characteristics which prevent companies realizing the benefits that could be obtain by their use. Respondents indicated that they invested significantly in customising many tools because the "off the shelf" packages were not satisfactory. It was found that software houses investigated more in the customisation of tools. The major criticisms of tools that were identified were:

- tools do not offer the possibility to use several methods in an integrated manner.
- the integration between tools for different software development phases is poor.
- tools often required a change to methods other than those used within the organisations.
- tools for project management are not integrated with tools for the other software development activities.
- the coupling of design data to the tool repository is weak.
- there are no multi-user application development possibilities, for communication between the CASE tool users.

Finally, it was found that, the large diversity of different tools was causing problems for software houses. Many respondents took the view that they should be able to work with any methods and tools that were required by their clients. However, this was a costly and difficult situation.

In Australia, the surveyed organisations agreed that currently CASE tools are costly and immature products. These two main factors were preventing the realisation of the benefits from automation. It was also noted that CASE tools are only as effective as the level of expertise of the staff using them. Experience of using immature CASE products during development projects has led to systems being developed that were immature and unstable.

In the United Kingdom, in addition to the major problem of cost, respondents identified a number of other problems with current tools. Figure 3 summarises these findings. Of these, poor documentation quality (as provided by the tool vendors) and the lack of multi-user development support were the most common. In identifying problems with tools and CASE in general respondents demanded to know CASE could be implemented without increasing the software backlog still further. It was reported that a considerable amount of effort and money had to be

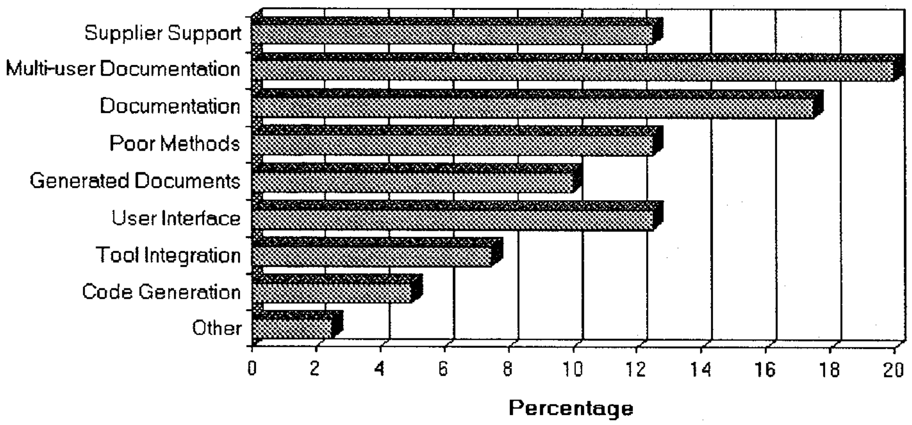

Figure 3 - Major problems with current CASE tools

committed but this had to be weighed against the impact on existing projects. Furthermore, respondents complained of the lack of management muscle to implement methods let alone CASE tools. Generally there was a considerable feeling of "sit back and wait" to determine the direction, standards and maturity of the CASE world.

In summary, tools in all countries seem to be considered immature. This, implies that they will mature and that tool users expect this too occur. However, current tools suffer from a number of distinct problems. Poor tool integration, lack of multi-user development facilities and poor method support were common problems in both the Netherlands and the United Kingdom.

2.6: Future Tool Requirements

Here we consider the future requirements which organisations believe CASE tools should satisfy. We also examine the features which, given the current state of tools, they would like to see in the next generation of products.

methods/tools	software houses	other organizations
need for a new generation of methods and tools	- yes: 20% - no: 80%	- yes: 0% - no: 100%
need for improvement of the current generation of methods and tools	- yes: 100% (but in particular for information planning and project control)	- yes: 20% (all methods and tools) - no: 80%

Figure 4 - Wishes and needs concerning methods and tools

In the Netherlands, the respondents wishes regarding the requirements for the next generation of CASE tools were interesting. Figure 4, summarises these findings. It was found that few respondents in organisations other than software houses wanted to adopt completely new methods and tools, nor indeed to make alterations/upgrades to their existing ones. They were just interested in trying to work effectively and efficiently with the tools and methods that they currently had. Respondents within software houses did not feel that there was a need for a completely new generation of methods and tools but they all felt that the current generation of methods and tools needed improvement. However, the actual requirements for the tool improvements were vague.

It was also found that super-tools supporting all phases of the software development life-cycle were not required. No organisation surveyed was interested in all phases and activities of the life-cycle.

In Australia, it was felt that CASE tools needed to be developed for Open Systems in order that tools could be easily selected for particular hardware and software requirements. Failure to produce new generations of tools for Open Systems would hinder the adoption and implementation of tools across organisations and hardware platforms.

Within the United Kingdom future tool requirements identified by both CASE and non-CASE users was also very vague. Some respondents wanted every aspect of current tools to be upgraded. However, CASE respondents were a little more selective with the need for multi-user development, full method support, high quality graphics and documentation accounted for 49% of their requirements. Non-CASE respondents were less specific. Some required every possible improvement be made with current tools, however, the majority tended towards improvement with the rear-end of the life-cycles tools supporting reverse engineering, and code generation etc. There was less requirements for front-end method support.

Thus, with regards to the future requirements of CASE there seem to be some major differences between requirements from each of the three countries. The Netherlands research indicates that while improvements in tools are needed they do not require full life-cycle support. In Australia the requirement of Open System tools and the increase in the maturity of current problems are key requirements. In the United Kingdom, multi-user systems, better method support and documentation were targeted areas requiring improvement. Although not specifically mentioned the requirements for Open Systems and multi-user development environment are similar. Thus, identification of proper standards and the development of more mature Open Systems tools would go some way to addressing these problems.

2.7: Use of Methods with Tools

It has long been agreed that tools should support methods. In this section we discuss the support of and the use of software methods within CASE tools and we consider whether CASE should support one or more particular methods.

Within the Netherlands, Van Reeken found that software houses especially, are forced to work with a variety of different methods and thus a variety of different tools were required to support these methods. The only area of the development life-cycle which particularly lacked method support was project control. Criticisms as regards methods were identified as: they require long learning periods, demand willingness from the developers and users, require users to gain considerable knowledge and experience concerning the method and require the adaption of new procedures.

In Australia both the surveyed organisations reported that the successful use of CASE required that a method was used which supported the traditional system development life-cycle.

In the United Kingdom both investigations indicated that the successful use of CASE was also recognised as depending on the implementation of a software development method. Figure 5, illustrates the use of methods and techniques by both Case and non-CASE survey respondents. Eighty one percent of respondents using CASE indicated that they used a recognised software development method. However, a surprising 20% claimed that they were not using any method but still used CASE products! This, could be accounted for by the use of lower-end life-cycle supporting tools. Almost 43% of respondents not using CASE claimed that they used no development methods. Of the remaining 57% who did use methods 39% used Structured Systems Analysis and Design Method (SSADM).

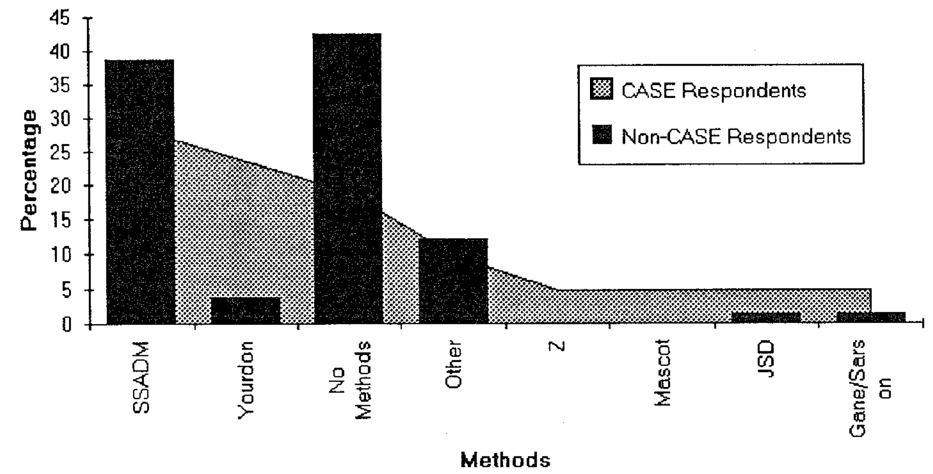

Figure 5 - Methods/Techniques Employed by Respondents.

Thus in summary, all three surveys identified the use of methods as being imperative. As to which method should be used this was unclear. However, SSADM was clearly identified amongst respondents as the leading method within the United Kingdom.

3: Summary Conclusions

This paper has compared and contrasted the results of three research surveys conducted within the United Kingdom, the Netherlands and Australia. These original surveys examined the empirical impact of CASE within commercial computing. We have found that many similar situations exist within the three countries, and this paper has highlighted the following key findings:

- The adoption and use of CASE is much lower across the sector than is often commonly reported.
- A major reason given for using CASE is to improve the overall software production process, especially in the area of control.
- It is currently very different to quantify overall gains in the areas of productivity, efficiency and quality arising from the use of CASE. Much more work clearly needs to be undertaken in this area before definite conclusions can be drawn. Currently, it would appear that any gains in one area are often offset by problems in another.
- The implementation of CASE is a costly business and should only be undertaken in a planned manner. Management support is essential if it is to succeed.
- CASE tools need to be flexible and capable of customisation and integration.
- Users are unsure on the future directions which CASE should take.
- The exact relationship between the use of CASE and the use of methods is far from clear. However, it would appear that since there are so many differing methods in use that CASE products need to remain very flexible.

The findings of the three investigations probably simply underline the still relatively immature nature of CASE. Much more needs to be done regarding the collection of actual case study data before more accurate conclusions can be drawn. However, it would appear that many users need much more information on and experience with CASE before they will be satisfied. Also perhaps rather than rushing ahead into the next generation of new products there should be a period of consolation and reconstruction of existing products.

4: References

[1] A.J. Van Reeken and J.J.M. Trienekens, "The Practical Importance of Methods and CASE Tools - Results of Empirical Research", Maastricht Economic Research Institute on Innovation and Technology, 1992.

[2] S.C. Stobart, J.B. Thompson and P. Smith, "Use, problems, benefits and future direction of computer-aided software engineering in United Kingdom", Information and Software Technology, Vol 33 no 9, November 1991.

[3] S.C. Stobart, J.B. Thompson and P. Smith, "A Survey of the Use of Computer-aided Software Engineering in the United Kingdom", Commercial Software Engineering Group, Occasional Paper No: 91-2, University of Sunderland, 1991.

[4] S. Majumdar and J.O.Jenkins, "Empirical Evaluation of CASE", Working Paper No. 93-1, City University, Department of Business Computing, 1993.

[5] D.R. Jeffery, G.C. Low and D. Bunker, "A Survey of Australian Organisations Using Computer Aided Software Engineering (CASE) Products", Experience Report, CASE 1992, Montreal, Canada, 1992.