16

CASE: Successful Implementation and Use

R. W. Smyth
School of Information Systems
Queensland University of Technology
PO Box 2434 Brisbane, Australia 4066
Tel.: +61 7 3864 2741
Fax: +61 7 3864 1507
Email: smyth@fit.qut.edu.au

Abstract

Computer Aided Software Engineering (CASE) offers system developers automated tools to replace manual methods. However, researchers and practitioners have reported a pattern of CASE being abandoned by many adopting organisations within two years of acquisition. Drawing on co-operation from a CASE vendor and a successful CASE user, this paper seeks to establish a basis for explaining how CASE success can be achieved. To investigate this issue, two main indicators of CASE success are initially identified. These indicators are: CASE utilisation, and Task-Technology Fit (TTF). TTF represents the extent to which there is a good match between the facilities of the particular CASE package, the development tasks carried out with the package, and the attributes of the developer using the package. Utilisation is measured by both the duration of use and the spread of use amongst eligible users in an organisation. Utilisation itself is shown to be influenced by organisational factors and, indirectly, by TTF. This initial framework is demonstrated using results from a successful CASE user, the Queensland Department of Natural Resources. Arising from analysis of the Department of Natural Resources case, a modified version of the framework is developed.

Keywords

CASE, implementation, use, success, case study

T. McMaster et al. (eds.), *Facilitating Technology Transfer through Partnership*
© IFIP International Federation for Information Processing 1997

INTRODUCTION

Although the theoretical benefits of CASE are strong, the twelve year history of commercial CASE packages could be seen as one of unfulfilled expectation. Evidence to date (e.g. Whybrow, 1989; Isoda et al, 1995; Rader et al, 1995) suggests that while some CASE users are experiencing significant benefits, others are less than satisfied and about one third abandon use of their CASE package within two years of initial acquisition. It is planned that a close examination of some successful CASE users will provide insights into the factors which enable them to prosper where others fail. Queensland's Department of Natural Resources was cited by management of LBMS, the supplier of the CASE package Systems Engineer, as an example of a successful CASE user.

The Queensland Department of Natural Resources (henceforth referred to as DNR) is a large government department administering a broad range of activities associated with the state's land, water and forests. Since the area of Queensland is several times that of Britain, this administration includes the gathering, management and processing of large amounts of data. Furthermore, many of the information systems services provided are very visible and important to public perceptions of governmental efficiency. Prominent among these is the computerised land titles system which records changes of title and allows real estate agents, lawyers and others to access title register details from their own offices. Similar access is available on land valuations and land sale prices. Politically sensitive information in the areas of environmental protection and native land title claims is also managed by the IT group in DNR.

With successive Queensland State Governments, the structure of what is now the DNR has undergone many changes, mostly involving the addition to it of groups previously part of other departments. In IT, this has posed challenges in integrating diverse computer based systems and in promoting consistent development standards across the department.

LITERATURE REVIEW

Computer Aided Software Engineering (CASE)

The term CASE is used to refer to a diverse range of software packages which share the common basic goal of providing automated support to information system developers. Within this broad grouping, a number of categories of CASE can be identified. Upper CASE packages are those which provide support for the early phases of the system development lifecycle viz. some or all of: Planning, Analysis, and Design. Lower CASE packages are those which support later phases of the lifecycle viz. Design, Database Development, and Code Generation. Integrated CASE refers to packages which address the lifecycle phases covered by

Upper CASE as well as those covered by Lower CASE. Component CASE involves the use together of Upper CASE and Lower CASE packages adhering to common standards. Through Component CASE, developers can receive assistance across the whole lifecycle, making use of a choice of tools. Systems Engineer, the package in this study, is an Upper CASE package with a common interface to a range of Lower CASE tools. Attributes and potential benefits and shortcomings of CASE are well described in the literature (McClure, 1989; Parkinson, 1991; Stone, 1993).

CASE Success

While CASE features have been widely covered, there has been limited evaluation of CASE and the factors contributing to its successful adoption. Much of the writing on CASE success is in the form of descriptions of lists of factors deemed to be determinants of success. Frequently, the lists are based on the basis of practitioner/ consultant perceptions (e.g. Parkinson, 1991). Wynekoop and Conger (1991), in a survey of CASE literature, highlight this neglect of CASE evaluation and the fact that much of the writing on CASE lacks underlying research rigour. Given this limited research base on CASE success, an appropriate strategy is to supplement what exists with theory from related fields. IS implementation, which now does have a solid theory base, is a relevant field, given that CASE implementation is a special form of IS implementation. Similarly, there is evidence that CASE adoption involves organisational changes comparable with those associated with the introduction of significant innovations, so that a consideration of innovation theory is warranted. Innovation adoption has been researched to the extent that a good body of theory has been built up there as well.

A piece of theory from IS implementation that offers promise in relation to CASE implementation is the concept of Task-Technology-Fit (TTF) as adapted by Goodhue and Thompson (1995). They describe TTF (p.218) as "the correspondence between task requirements, individual abilities, and the functionality of the technology". Using data from 600 respondents, Goodhue and Thompson validated the TTF construct. They also showed that as predictors of IS implementation success "both TTF and utilization must be included" (p.228).

In a separate paper, Goodhue (1995) shows that users can successfully evaluate TTF. This is important in attempting to measure the TTF construct.

A MODEL OF CASE SUCCESS

At the beginning of this study, two main factors were initially identified as indicators of CASE success. These draw on the literature on IS implementation and on innovation adoption. They also take account of literature on CASE success, and the results of an earlier case study examining CASE success factors

(Smyth, 1997). These two factors are (1) Task-Technology-Fit (TTF) as it applies to CASE and (2) CASE utilisation. A model of CASE success is shown in Figure 1. This shows TTF as resulting from the interaction between the CASE package (here, Systems Engineer), the individual developer's knowledge, skills and attitude to CASE, and the development tasks attempted using the CASE package. TTF then influences thc developers' perceptions of the usefulness of CASE, and their view of the relative value of CASE. These factors, in turn, influence the utilisation of CASE by the developers. Also influencing CASE utilisation are a variety of potential organisational factors which might include senior management support, the presence of an organisational champion for CASE, personnel policies, and organisation politics.

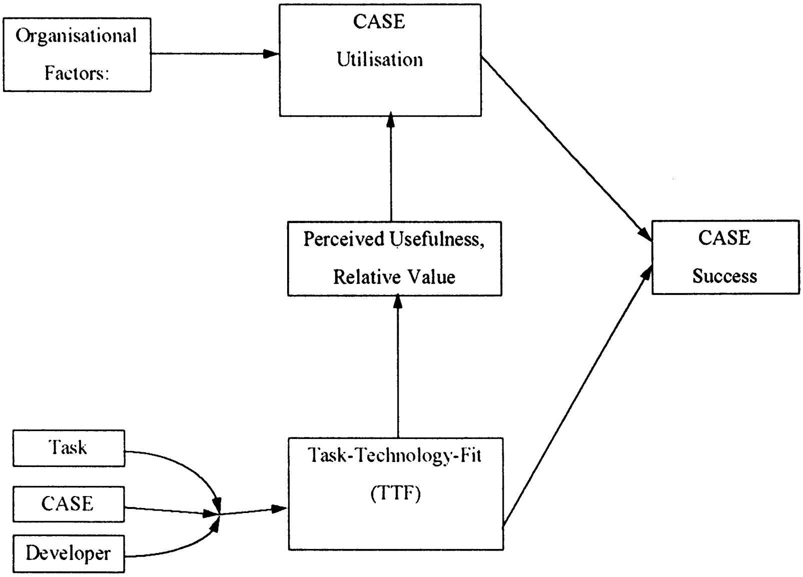

Figure 1 Model of CASE Success.

THE RESEARCH METHOD

The Case Study Approach

This paper is based on a single case study as a form of explanatory research. An earlier case study (Smyth, 1996) has been carried out with an exploratory objective. From the exploratory case study and related research, a tentative model of CASE success is proposed. This model is shown in Figure 1. The model has been used to guide the conduct of the case study. This idea of using a theory based model as a means of providing focus for the case study is consistent with Yin's view (1994, p.28) that "theory development prior to the collection of any case study data is an essential step in doing case studies". It should be noted, however, that this approach is at odds with the "grounded theory" view (Glaser and Strauss, 1967) which argues in favour of iterative induction from the field data itself to the development of propositions or hypotheses. Here, with awareness of the potential richness of case study data and a willingness to follow up lines of enquiry extending beyond the confines of the initial framework, it is felt that the danger of discarding potentially important data has been reduced.

The findings at DNR are compared with the proposed Model of CASE Success. On the basis of observations at DNR and consideration of relevant theory, the initial model is revised. Care is taken to ensure that the integrity of the revised model is not jeopardised by the changes prompted by specific observations. The overriding concern is that the model which evolves is not only consistent with observed facts, but is also respectful of established theory and in a form accessible to practitioners.

Case Study Protocol

A case study protocol was developed prior to data collection. The format of the protocol conforms basically with that recommended by Yin (1994, pp.63-74). As such, it outlines the model, with justification for it, a list of operational measures for each variable identified in the model (see Table 1), validity and reliability provisions, an outline of the data collection methods, and a list of specific questions to be addressed in data collection. Not only was the protocol useful in providing focus for the case study but it is a significant contributor to the reliability of the study. There is sufficient detail in the protocol about the model tested and the methods used that the study could be readily repeated.

The CASE Utilisation Construct

Research in implementation and innovation show two widely used utilisation dimensions. The two dimensions are: persistence of use, and extent of use in the organisation. Persistence of use is a component of utilisation that relates to the basic idea of continuing use over a sustained period of time. CASE research suggests (Selamat et al, 1994; Isoda et al 1995; Rader et al, 1995) that non-use is a common phenomenon one or two years after initial CASE adoption. Extent of use equates to Rogers' (1983) idea of the degree of diffusion. This dimension of utilisation is widely used in studies of IS implementation (Schewe, 1976; Robey, 1979; Raymond, 1985). McChesney & Glass (1993) use it in a study of post-implementation management of CASE. In this DNR study, persistence of use and extent of use are employed as the two dimensions of CASE utilisation.

The Task-Technology-Fit Construct

As outlined earlier in this paper, Goodhue (1995) has shown that user evaluation of TTF is an accurate representation of TTF. In this study, the evaluation by IT development staff of TTF as it relates to Systems Engineer's use at DNR is assumed to be a sound representation of TTF. Where in Goodhue's use of the TTF concept task differences relate predominantly to differences in complexity, in the context of CASE adoption other differences are seen to predominate.

Measures of the Variables of Interest

In this case study, measures were determined for each of the main variables identifiable from the proposed model of CASE success. These measures are used to underpin the rich data collected from DNR to evaluate the model. The measures are shown in Table 1.

Table 1 Measures for Variables in the Proposed CASE Success Model

Variable	Operational Measure
Organisational factor: Strategy for CASE introduction	Evidence of a plan outlining at least: Planned phases of introduction, personnel responsibilities.
Organisational factor: Extensive training of development staff	Dollars spent on training on the CASE package, 'extensive' implying at least $0.20 for every $1 spent on the software purchase
Organisational factor: Senior management commitment	Presence of a senior manager on a steering committee for CASE introduction, and/or assessment by IS management of commitment by senior managers (7-point scale)
CASE Utilisation - Persistence of use	Evidence of continued use of the CASE package at least 1 year after introduction
CASE Utilisation-Extent of use	% of 'eligible' developers using CASE; range of Systems Engineer features used
Perceived usefulness of SE	Developer evaluation of the extent of benefit from using Systems Engineer (7-point scale)
Relative value of SE	Developer evaluation of benefit from SE relative to the economic and other costs in its adoption (7-point scale)
Task-technology-Fit	Developer evaluation of the match between the CASE package, the tasks attempted with it, and the skills, knowledge and attitudes of the developer (7-point scale)

METHODS OF DATA COLLECTION

A range of staff members from DNR's Information Technology Branch was interviewed. This included: the Senior Computer Systems Officer in Database Administration, as the person with operational responsibility for Systems Engineer; the Director of Information Technology Services for DNR, as the senior manager with overall responsibility for adoption and use of Systems Engineer; the IT Quality and Customer Services Officer, as the author of the department's IT Procedures Manual; and various DNR Information Technology development staff, as the people in the Department with greatest operational experience of Systems Engineer. While the interviews focused on factors and interactions represented in the model to be tested, the format of each was sufficiently open to permit the collection of important data unanticipated by the research framework.

Other forms of data collection were employed to complement the interviews and to provide a basis for triangulation of evidence. In particular, the Department's Quality Manual, Procedures Manual, and Standards Manual, together with the recently implemented online developer guidelines, provided data on the use of Systems Engineer and the development context in which it is used. Some base data gained by interview and documentation from LBMS, the CASE supplier. was also used to expand and check against data gathered from the Department.

EMPIRICAL FINDINGS FROM THE DNR CASE STUDY

Task-Technology-Fit (TTF)

There is some variation among DNR respondents with regard to the TTF although the overall view is favourable. The variations can be largely attributed to the multifaceted nature of system development. Systems Engineer in the DNR setting suits well the need to document user requirements and new system specifications in a way that can be shared among developers and in a way that facilitates conformity to standards and good version control. The match between the CASE package and the development task is seen as less favourable in specific aspects of the system design process and in the links to database development and code generation. In these aspects. DNR developers view the capacity of Systems Engineer to meet their needs as limited. This is particularly true with regard to the idea of Component CASE where Systems Engineer has been created to interface readily with specific back end products so that an Integrated CASE toolset is available. The perception of DNR developers is of inadequate integration causing them to limit their use of CASE to Upper CASE functions.

It should be noted that the proposed Model of CASE Success does not provide for the observation from DNR that the perception of TTF is influenced by initial expectations of the CASE package. Where initial expectations were for the CASE package to provide improved quality and improved productivity, the perception of TTF is lower. To better explain CASE success, the model should be refined to show Initial Expectation as a concept affecting the perception of the task and , hence, affecting TTF.

The strength of perceived TTF in the Department was not influenced by the complexity of the projects to be supported by Systems Engineer. Although most experience with Systems Engineer at DNR was with large, complex projects, IS staff felt that the facilities provided by Systems Engineer matched up well with the needs of all but the most trivial of projects. Sound, standard documentation for communication with other developers and with end-users is seen as important regardless of the size and complexity of the project. Likewise, the development of a project repository to facilitate good version control and to accommodate change during and after system development is also seen as desirable for all projects. In this regard, the application of TTF to CASE implementation is different from that used by Goodhue and Thompson (1995) where the main variation in task relates to level of complexity.

Those DNR development staff who have a perception of very strong TTF tend to be ones who are impressed by the capacity of Systems Engineer to improve the quality of the system development process. The IT Quality and Customer Services Officer, for instance, sees Systems Engineer as well suited to DNR's development tasks. Those who nominate a less strong TTF rating, acknowledge this good match with quality needs but see limitations in the capacity of CASE improving productivity in system development. Although there are exceptions (e.g. Finlay & Mitchell, 1994), the DNR situation is mirrored in most reports on CASE use (e.g. Rader et al, 1995). These generally report CASE successes in addressing Quality issues but limited outcomes in assisting Productivity in systems development.

To raise the general TTF rating for CASE adoption would seem to require improvement in CASE packages to enable them to provide a higher level of assistance in improving system development productivity. This would also imply that the tasks supported by CASE for most organisations would expand to include a greater part of the system development lifecycle. Given the size and complexity of existing CASE packages such as Systems Engineer, further increases in the functionality of CASE packages will increase the extent of training and support needed by users of these packages.

Perceived Usefulness, and Relative Value

Seddon and Kiew (1994) "..found that Usefulness is concerned only with the future benefits of performing some task. Costs are much less important" (p.103). Using this definition, then, perceived usefulness of CASE is the perception of developers that using the CASE package would lead to future benefits being realised. Information Systems staff at DNR rate Systems Engineer as Useful. As suggested in the CASE Success Model proposed, the Perceived Usefulness of CASE in DNR reflects the perception of TTF. Those who perceive the best match between the development task, Systems Engineer's capabilities, and the attributes of DNR developers, have a view of Systems Engineer as very useful because of its potential to provide significant improvement in the quality of the development process. Where the TTF is rated lower because of initial expectation that the package should assist in productivity improvement as well as quality improvement, the perception of usefulness is reduced.

A significant observation at DNR in relation to this concept of Usefulness is concern, particularly by their IS management, about the contribution that Systems Engineer might be able to make in a newly emerging development environment. DNR is moving to more client-server development, it is looking to an intranet as the basis for improved groupwork and it is watching with interest the possible benefits of object oriented analysis and design. In the spirit of Seddon and Kiew's view of usefulness as relating to "future benefits", DNR staff are reassessing Systems Engineer's usefulness to them. Systems Engineer specifically targets client-server development support although DNR report some operational shortcomings in this. Although the intranet will provide alternative means for collaboration among developers, it should not lessen the benefits of good multi-user support provided by Systems Engineer. LBMS has not announced an intention to include support for object-oriented analysis and design in Systems Engineer.

The Usefulness construct as envisaged in the initial model of CASE success was explored from the viewpoint of the benefit to be derived from its use in the organisation. In the light of the facts from DNR, future considerations of CASE usefulness should explicitly address use in possible future environments.

The concept of Relative Value is related to but distinct from Usefulness. Where usefulness expressly excludes the matter of costs, Relative Value requires the users to evaluate benefits relative to the costs. In the context of CASE adoption, the costs include the acquisition and maintenance costs of the software as well as the organisational costs in implementation and use. Again, at DNR the evaluation of the Relative Value of Systems Engineer paralleled the assessments of TTF. Where the initial view of the tasks to be addressed was limited to quality-related matters, the perception of TTF and then of Relative Value is highest. In summary of the prevailing view on Relative Value at DNR, the Director of IT Services remarked, "We've got our money's worth from Systems Engineer". Again though, considerations of future directions in system development tended to cause DNR respondents to question their Relative Value assessments.

Organisational Factors Influencing CASE Utilisation

At DNR, a Quality Policy is in place. This policy has significant impact on the utilisation of Systems Engineer. The Department's policy is related to the previous State Government's policy on quality. Under the government which held power in Queensland for some eight years up to March, 1996, it had become mandatory for suppliers to the government to be quality certified. State government departments were encouraged to put in place Quality Policies. In this climate, DNR had instituted its policy and at the time of change of government had been on the point of seeking certification to ISO9000 and to AS3563, the Australian standard on software development management. Although certification had not proceeded, DNR has maintained its commitment to quality in system development. Against this background, procedures are prescribed which require use of Systems Engineer on development projects, including those which are outsourced. Iivari (1996) found that a requirement to use CASE was one of only two predictors of CASE usage in an organisation. Although this is intuitively obvious, the issue of voluntariness in use would appear to be important at DNR. With external contractors, there have been some problems for DNR in ensuring the existence of adequate skills in the use of Systems Engineer. In light of the impending relaxation of Quality requirements by the state government, retention of the existing Quality Policy, and hence the extent of future use of Systems Engineer, will be dependent on the extent to which a Quality Culture has been internalised at DNR.

Top management support exists for the utilisation of Systems Engineer at DNR. Management support was the second of the two predictors of CASE usage found by Iivari. Systems Engineer's use is consistent with continuing senior management support for quality processes. The support is shown in provision of adequate resources for acquisition and maintenance of Systems Engineer licences, and for comprehensive training of development staff.

The utilisation of Systems Engineer at DNR takes place in the absence of a champion for its use. Both the Database Administration manager, who has operational responsibility for its use, and the Director of IT Services, who has overall responsibility for Systems Engineer's utilisation at DNR, acknowledge the contribution Systems Engineer makes to the quality of the development process. Yet both question its relative value in a changing application development setting.

No overt political factors of the kind described by Markus (1983) and others appear to influence current utilisation of CASE at DNR. However, with the likelihood of a relaxation of Government's attitude to Quality, this could become a factor in the approaches shown to Systems Engineer by staff in the Information Technology Branch.

It is acknowledged that organisational influence on CASE utilisation could take many forms apart from those observed at DNR. Other writers on CASE use (e.g. McClure, 1989; Parkinson, 1991; Smyth, 1997) have described a range of organisational factors which have the potential to influence (positively, or negatively) the utilisation of CASE in that organisation.

CASE Utilisation

DNR meets the persistence of CASE use criterion, having used Systems Engineer consistently over a six year period. DNR, in fact, had used Systems Engineer's predecessor, Auto-Mate Plus. It had also during the early 1990s used another CASE package, IEW, on an enterprise modelling project. This use of IEW had been concurrent with use of Systems Engineer on other projects. DNR had started with one copy of Systems Engineer Release 1 in 1990 and had increased the number of licences in stages, reaching 25 prior to the case study. It is notable, however, that although the 25 licences were in force at the time of the case study, about half of them were not being actively used at that time. This reflects the completion then of some large development projects.

The extent of CASE use at DNR is great when measured by the percentage of eligible developers who use Systems Engineer. During the analysis and design phases, all analysts and all analyst/programmers doing this work use Systems Engineer. The terms of the procedures to be followed and outputs to be produced, as detailed in the Branch Procedures Manual, require use of Systems Engineer. A great many of the features of Systems Engineer are regularly used by DNR staff. All of the support features for Structured Analysis are regularly used as are many of the structured design features.

CASE Success

According to the proposed model of CASE success, DNR would be deemed strongly successful in its adoption of CASE. The Task-Technology-Fit observed for DNR is quite strong. The utilisation as related to the pre-established measures is very strong.

However, this explanation does not adequately reflect elements of disquiet at DNR, reported in the body of this discussion, about the role of Systems Engineer. To some extent, these concerns show through in the TTF observation although the essence of this perception emerged in discussions extending beyond the framework initially set. (This is an advantage of the case study method). An extension of the original model is needed to accommodate the shortcoming detected by this data from DNR. A more adequate depiction of CASE success, taking into account this finding at DNR, might be achieved by including User Satisfaction, a widely used construct in implementation research (Ives et al, 1983). User Satisfaction with CASE as measured among the IS staff would be considered in conjunction with TTF and Utilisation to determine CASE success.

A REVISED MODEL OF CASE SUCCESS

The initially proposed model of CASE success, as shown in Figure 1, was compiled on the basis of an earlier, explanatory case study and existing formal theory from the related fields of implementation and innovation. Based on observed facts in the DNR case and further reference to established theory, a revised model of CASE success has been developed. This revised model is shown in Figure 2. Eisenhardt (1989) supports the approach used here, stating: "Overall, tying the emergent theory to existing literature enhances the internal validity, generalizability, and theoretical level of theory building from case study research" (p.545).

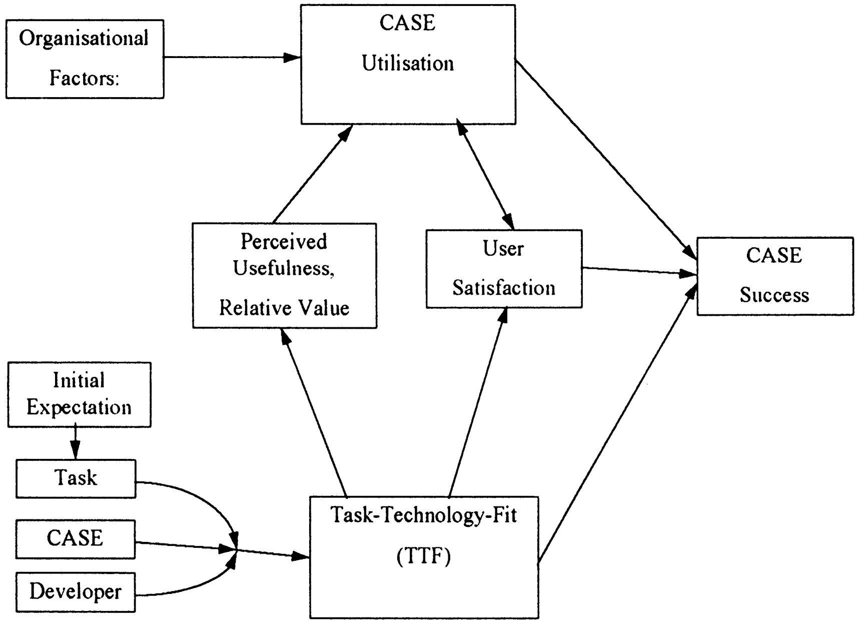

Figure 2 Revised CASE Success Model.

RECOMMENDATIONS FOR PRACTITIONERS

The theoretical framework developed here and shown in Figure 2 has important implications for Information Systems Managers and for CASE vendors. The framework relates specifically to the real-world adoption and use of CASE, having emerged from a study of routine use of a CASE package. The framework is applicable across a range of situations because it is tied to established theory in implementation and innovation. Most importantly, it is readily intelligible to practitioners.

For IS managers planning to acquire and install a CASE package, there are a number of recommendations. Prior to the selection of a CASE package, there is advantage in clarifying management's expectations of a package. This could initially be formalised in the selection criteria and/or the Request for Proposal. After selection, the proposed scope of the package's use should be made clear in the training program. To have conflicting views on the expectations of CASE in the organisation is to have diverse perceptions of the TTF and, hence, to impact the success of CASE in the organisation.

The concept of TTF provides some guidelines for management to follow in CASE selection. The selection process should, firstly, seek a good match between the development support tasks identified as the target for CASE and the package selected. This implies that the organisation should clarify the development methodology or methodologies they feel best suited to their medium-term future. It also requires that the organisation determine the proposed working mode for their developers e.g. collaboration and joint design, and the system architecture/s for which they will be developing e.g. client-server. As was pointed out in this case study, the job of matching the package selected with the tasks to be performed also requires that management evaluate emerging trends as well as current practices. In taking these trends into account, the selection team will need to assess the capacity of each CASE contender to accommodate the likely changes and will need to look at the track record of each vendor in having kept pace with or anticipated change via various versions of the package in the past. The developers should also be assessed in terms of how they will interact with the prospective package and the tasks to be completed with it. Knowledge of CASE from their formal education or prior employment, attitudes to replacing well established skills with new ones, competence in proposed methodologies - all of these will contribute to the TTF for any CASE package under consideration.

Organisational factors are shown to influence the utilisation of CASE. This study has indicated that the specific factors which can have greatest impact (positive or negative) will vary across organisations. The research on interaction between organisation and technology is replete with contradiction (Robey, 1995). What is established, though, is that organisational factors should not be ignored. The IT manager contemplating the adoption of CASE should scan the factors likely to promote or inhibit for his or her organisation. Are there power groups who would have an interest in seeing CASE succeed/fail? Is there an obvious champion for CASE? What is the attitude of top management? Is there an organisation culture that could work for or against successful adoption and use? Where threats are detected, the manager has the choice of deferring adoption or seeking to neutralise the threats. Where positive factors are noted, these should be exploited and promoted.

There are implications for CASE vendors from this research. First is the obvious need to keep packages abreast of change and, hence, in line with perceptions of good TTF. Refinement of CASE packages to provide support, in a form acceptable to developers, through all phases of the development lifecycle is another means of improving TTF. Beyond this, is the need to market the package in such a way as to paint a strategic picture for an individual client. This study has shown that there is a need for the vendor not only to convince the client of the suitability for current needs but also to show how the tools will fit into the client's longer-term scenario of system development.

RECOMMENDATIONS FOR FUTURE RESEARCH

It is suggested that the revised CASE success model be used as the starting point for further case study investigations. Retention of a case study approach may increase the likelihood of detecting subtle interactions which might otherwise go undetected given the limited quantity of prior CASE research. The comparison of further case study data with the model should permit progressive improvement and refinement of it. Particular emphasis should be given to the Task-Technology-Fit construct. It may be that by modifying TTF to take account of anticipated changes in an organisation's approach to development, TTF will be a more powerful indicator of success. This might permit adequate explanation of CASE success without the need for measurement of User Satisfaction.

With a stable explanatory model developed from multiple case studies, the research could be further extended by testing the model through development of a suitable instrument for use in a survey of CASE users. The collection of and analysis of such quantitative data could complement and extend (Gable, 1994) the understandings of CASE success factors gained from the case studies.

REFERENCES

Eisenhardt. K. M. (1989) Building Theories from Case Study Research. *Academy of Management Review*, V.14, No. 4, Oct., 532-550.

Finlay. P. N. and Mitchell. A. C. (1994) Perceptions of the Benefits from the Introduction of CASE: An Empirical Study, *MIS Quarterly*, Dec., 353-370.

Gable, G. G. (1994) Integrating case study and survey research methods: an example in information systems, *European Journal of Information Systems*, Vol. 3, No. 2, 112-126.

Glaser, B. and Strauss, A. (1967) *The Discovery of Grounded Theory*, Aldine Publishing.

Goodhue, D. L. (1995) Understanding User Evaluations of Information Systems, *Management Science*, Vol. 41, No. 12, December, 1827-1844.

Goodhue, D. L. and Thompson, R. L. (1995) Task-Technology Fit and Individual Performance, *MIS Quarterly*, June, 213-235.

Iivari, J. (1996) Why Are CASE Tools Not Used?, *Communications of the ACM*, Vol. 39, No. 10, October, 94-103.

Isoda, S., Yamamoto, S., Kuroki, H. and Oka, A. (1995) Evaluation and Introduction of the Structured Methodology and a CASE Tool, *Journal of Systems Software*, no. 28, 49-58.

Ives, B., Olson, M. H. and Baroudi, J. J. (1983) The Measurement of User Information Satisfaction, *Communications of the ACM*, V.26, Oct., 785-793.

Markus, M. L. (1983) Power Politics and MIS Implementation, *Communications of the ACM*, June, V.26, No. 6, 430-444.

McChesney, I. R. and Glass D. (1993) Post-implementation management of CASE Methodology, *European Journal of Information Systems*, V.2, No. 3, 201-209.

McClure, C. (1989) *CASE Is Software Automation*, Prentice-Hall.

Orlikowski, W. J. (1993) CASE Tools as Organizational Change: Investigating Incremental and Radical Changes in Systems Development, *MIS Quarterly*, September, 309-340.

Parkinson, J. (1991) *Making CASE Work*, NCC Blackwell.

Rader, J., Brown, A. W. and Morris, E. J. (1995) Operational Use of CASE Integration: An Investigation of the State of the Practice, *Journal of Systems Software*, v. 28, 59-68.

Raymond, Louis (1987) Validating and Applying User Satisfaction as a Measure of MIS Success in Small Organisations, *Information & Management*, No. 12, 173-179.

Robey, D. (1979) User Attitudes and Management Information System Use, *Academy of Management Journal*, September, 527-538.

Robey, D. (1995) Theories That Explain Contradiction: Accounting for the Contradictory Organizational Consequences of Information Technology, *Proceedings of 16th International Conference on Information Systems*, Dec., 55-63.

Rogers, E. M. (1983) *Diffusion of Innovations*, Free Press of Glencoe.

Schewe, C. D. (1976) The Management Information Systems User: An Exploratory Behavioral Analysis, *Academy of Management Journal*, Dec., 577-589.

Seddon P. and Kiew, M.-Y. (1994) A Partial Test and Development of the DeLone and McLean Model of IS Success, *Proceedings of the 15th International Conference on Information Systems*, Dec., 99-110.

Selamat, M. H., Choong, C. Y. and Othman, A. T. (1994) Non-use phenomenon of CASE Tools: Malaysian Experience, *Information and Software Technology*, v. 36, no. 9, 531-537.

Smyth, R. W. (1997) CASE Success at Bank of Queensland, *Proceedings of 1997 Pacific-Asia Conference on Information Systems*, April.

Stone, J. (1993) *Inside ADW and IEF: The Promise and Reality of CASE*, McGraw-Hill.

Whybrow, M. (1989) Substantiate the CASE Mirage, *Infomatics*, June, 6-10.

Wynekoop, J. L. and Conger, S. A. (1991) A Review of Computer Aided Software Engineering Research Methods, in. H.-E. Nissen, H. K. Klein and R. Hirscheim (eds.) *Information Systems Research: Contemporary Approaches and Emergent Traditions*, Elsevier Science, 301-325.

Yin, R. K. (1994) *Case Study Research: Design and Methods* (2nd ed.), Sage.

BIOGRAPHY

Bob Smyth is Senior Lecturer in Information Systems in the School of Information Systems at Queensland University of Technology (QUT), Australia. He is also Assistant Dean (Postgraduate) in the Faculty of Information Technology at QUT. Prior to joining QUT, he worked as a Systems Engineer for IBM Australia.