System Development in the New Millennium: An Australian Perspective on OO Adoption

Ee Kuan Low[1]
Angèle L M Cavaye[2]
Aileen Cater-Steel[1]

[1]Department of Information Systems
University of Southern Queensland
Toowoomba, Queensland, Australia
Email: eekuanlow@usq.edu.au

[2]Graduate College of Management
Southern Cross University
Coolangatta Campus, Queensland, Australia

Abstract

In recent years there has been increased interest in object-oriented (OO) methods as a quicker, more comprehensive and viable alternative to traditional data- and process-oriented methods. This paper presents the first phase of exploratory research into understanding the issues relating to making the decision to adopt OO, the actual process of OO adoption and evaluation of OO adoption within the Australian context. A case study was conducted involving a large government organisation. The findings from the case study provide a fascinating insight into the realities of the OO adoption process.

Keywords

Change management, exploratory research, information technology adoption, IS development methodologies, object-oriented approach, case study.

INTRODUCTION

The turn of the century and beyond is perceived and accepted by most writers as an age of continuous change. These changes are especially evident in the areas of business management and information systems. As businesses rely more and more on computing technologies and information systems development to cope with changing requirements, it is acknowledged that the traditional approaches to system development are inadequate in coping with these changes (Taylor 1992; Brown 1997; Sultan & Chan 2000).

In order to overcome limitations of existing traditional systems development methodologies (SDMs), IS practitioners are interested in adopting better and more effective SDMs. A major candidate as a new methodology is OO - the methodologies and techniques based on object orientation. OO has been in existence for nearly 30 years, but it is only with recent developments in programming languages and supporting hardware that it is currently seen as a promising viable method as an alternative to traditional methodologies (Johnson & Hardgrave 1999a).

There is limited knowledge of the factors that influence the decision-making process when organisations consider OO adoption, limited knowledge of the ways in which the system development process needs to be adapted for OO development, and limited knowledge about the way in which OO methods can most effectively be introduced and evaluated (Dick & Rouse 1994; Fichman & Kemerer 1997). Despite the general lack of knowledge, many organisations are seriously considering OO adoption, are in the process of adopting OO, have

adopted OO in some form, or are now evaluating the productivity enhancements of OO (Barnard 1998). Clearly, there is a need for research to address the lack of knowledge and to enable the development of guidelines for improved practice.

This paper starts with a discussion of OO's promise as the appropriate SDM for the new millenium. This is followed by the presentation of a research framework that could be used to investigate OO adoption. Findings from the case study are discussed and finally, issues for further research are put forth.

OO: AN APPROPRIATE SDM FOR THE NEW MILLENNIUM

Advocates of OO assert that OO shows much promise as an appropriate SDM for the new millennium. The reasons for an increased interest in OO systems development are manifold and are now discussed.

Utilises a 'complete' methodology. OO is a methodology that provides a step-by-step process that provides segregation of phases to reduce unnecessary complexity (Federowitz & Villeneuve 1999). At the same time, there is ease of transition between the phases of analysis to design to implementation. Because of this, it is advocated as a complete system development methodology that concerns analysis and design and not just programming and implementation.

Handles systems of arbitrary size and complexity. The OO data model is a superset of all the preceding generations of data models and hence, is able to cope with legacy systems. Also, its all-encompassing data model is able to represent data models of any arbitrary complexity and size (Taylor 1992; Brown 1997). OO's encapsulation of related data and procedures is an effective tool for hiding or revealing complexity, allowing for freedom of expressing differing degrees of complexity in a business system.

Provides flexibility in dynamic and turbulent environments. OO fares well on the flexibility criterion (Taylor 1992; Johnson & Hardgrave 1999b). If designed well, the high degree of modularity enforced in OO methodology allows for changes to be implemented in parts of the business without affecting other parts of the business. Reusability of business objects through its inheritance hierarchy decreases the time needed to develop business systems (Berard 1996).

Supports integrated organisational solutions. OO combines the effectiveness of both process and data methodologies. With its ease in integration with organisational structures via object relationships, OO works to move system development from one-off solutions to integrated organisational solutions. OO assists in viewing and modelling the organisation in a 'real-world' context – the modelling of people interacting with business objects to produce meaningful reports for effective decision-making (Brown 1997).

Enforces standards in a methodology. Differing views are offered here. The optimists (Taylor 1992) suggest that OO methodologies are ready for prime time and are no longer a promise, but a reality. After all, the OO programming component has been around for more than 30 years. On the other hand, there are numerous variants of the OO methodology. Pessimists and realists (Adhikari 1996; Fichman & Kemerer 1997) assert that standards in OO methodologies are as yet non-existent and that such standards are pre-requisites for widespread adoption of OO.

Benefits of OO Approach

OO SDMs have received attention in information systems development due to the advantages over traditional approaches claimed by OO proponents. Numerous articles based on practical experience and empirical work have detailed the promises that drive organisations to jump on

the OO bandwagon (Taylor 1992; Dick & Rouse 1994). The main benefits identified in the literature are as follows.

Increased reuse in system development. OO concepts of inheritance and polymorphism encourage the reuse of existing objects. Component-based development is performed at a fraction of the time and cost of traditional system development (Taylor 1992; Meyer 1987; Hantos & Joseph 1997).

Greater resilience to changes in the business. Low coupling and high cohesion within and between objects increase the maintainability of OO systems making OO systems more resilient to changes in the internal and external environments. OO systems are generally easier to modify and maintain than systems developed using the traditional SDMs (Henry & Humphrey 1992; Taylor 1992; Johnson & Hardgrave 1999b).

Faster system development. The notion of reusability and standard methodology notations, when implemented correctly, create highly reusable OO objects. Component-based system development infers a quicker and faster way to develop systems (Taft 1995; Hantos & Joseph 1997). The promise of OO to eradicate problems of the mythical man-month of IS projects has contributed in a major way to its acceptance as 'the' SDM of the millennium.

Better support for the creation of graphical user interfaces. The importance of GUIs in system development can no longer be ignored as most business systems today consider graphics as not only appropriate, but essential. Most OO-based techniques, tools and support tools are GUI-based to better support the creation of the GUI component of a system (Dick & Rouse 1994).

Production of higher quality systems. Because OO systems are largely constructed from existing, proven components, the methodology typically yields a system that is assembled from high quality, accurate and error-free objects. The end-result is a system that is of higher quality than those developed from scratch, as in traditional SDMs (Fichman & Kemerer 1992; Taylor 1992; Johnson & Hardgrave 1999b).

Risks Inherent in OO Adoption

Although OO provides many exciting opportunities and benefits to a potential adopter, projects contemplating the use of OO carry some inherent costs (Taylor 1992; Hantos & Joseph 1997). The main costs and risks identified in the literature are as follows.

Technology investment. Most IS projects face critical decisions in the use of new technology (Hantos & Joseph 1997). OO projects are inherently based on new technology as adequate technology is needed to support the activities of OO system development. Perceived risks include the incompatibility of OO with legacy system development and maintenance (Taylor 1992), suitability of methodology to suit the business philosophy (Adhikari 1996), vendor support and the maturity of OO SDMs (Fichman & Kemerer 1993).

Staff training and experience. A sufficient number of experienced people are required among managers, system architects, and developers for the overall effectiveness of the OO development team (Hantos & Joseph 1997). There are two main overriding themes in the literature of OO adoption with regards to the adoption costs related to the development team: paradigm shifts in mindset (Hantos & Joseph 1997); and the learning curve to arrive at a satisfactory level of OO practice (Attewell 1992). Unskilled development teams may render the OO system to be a worse system than if traditional SDMs were used (Johnson & Hardgrave 1999b).

Management education and support. Management style and technique will need to be reviewed when adopting OO. Managers must realise the commitment required of them to support OO adoption. Three main areas would concern business managers when adopting

OO. Firstly, reward structures must be re-assessed to encourage code reuse (Taylor 1992). For example, lines of code would no longer suffice as measurement of programmer productivity. Secondly, management must be prepared for new employee positions and relationships to emerge, including a designated 'objects librarian' to index, manage, instigate, promote and facilitate re-use and a new object consumer-producer relationship (Taylor 1992; Classe 1995). Finally, these changes that occur within the development team and the organisation as a whole, require careful management.

Commitment to reusability. One of the ultimate goals of adopting OO SDMs is the increase in the reusability of software components (Taylor 1992; Fichman & Kemerer 1992, 1993; Brown 1997). Even so, adopting OO does not guarantee reusability. Reusability is not a direct result of using OO SDMs, it is more a result of properly managing the OO system development process. The OO development team must have a commitment to reusability. This means that developers should design for future uses, not just for the problem at hand. The object consumer must trust the abilities of the object producer to construct highly reliable and generic reusable objects (Taylor 1992; Fichman & Kemerer 1997).

Table 1 summarises the main risks associated with OO adoption and provides a number of mitigation strategies suggested by the OO adoption literature.

Risks	**Mitigation strategies**
Incompatibility of technology	Pilot new technologies on a small scale before adopting them on large projects.
Suitability of methodology Maturity of OO SDMs	Use S-curve analysis to predict the future usage and popularity of a methodology. Also, OO standards-making organisation such as OMG's recommendations on specific methodologies could be a starting point in the selection process.
Continuous vendor support	Evaluate vendors based on its market status, product features and product support terms.
Paradigm shifts in mindset	Provide foundation-level training (OO concepts) up front before project-specific training (OO implementation language).
Learning curve	Create an environment to learn incrementally and through customised mentoring programs.
Outmoded reward structures Enforcing re-use	Re-evaluate reward structures with a focus on reusability.
New structure of employee relationships Paradox: producer vs. consumer	Take personality trait of employees into consideration in the recruitment process. For existing staff, staff education and reward structure may assist in the transitory phase. Also, there may exist a need to re-evaluate organisational structure, hierarchy and relationships.
Change management	Facilitate close-knit employer-employee relationships through open communication channels.

Table 1: Perceived costs and mitigation strategies (Adapted from Taylor 1992; Hantos & Joseph 1997; Johnson & Hardgrave 1999a, 1999b)

Making the decision to adopt. Making the decision to adopt OO will require in-depth analysis and evaluation of its benefits and costs. Costs and risks are associated with any type of adoption but decision-makers have to be assured that these risks can be mitigated, or at the very least, be reduced to a satisfactory level, to reap the intended benefits of adoption. The benefits of changing from traditional to OO methodologies must exceed the costs. Most importantly, there is a need for a collaborative decision to be made by both development and management teams to see if the benefits of such an investment can be realised within the time- and cost-frame of the system development project.

CURRENT STATE OF OO ADOPTION

OO has found increased acceptance and use in a number of commercial software development organisations, consultancies, business organisations and universities (Reed 1995). Research

centres have been established in order to understand and properly manage this paradigm shift in system development.

Adoption and Use of OO

In evaluating current OO adoption and predicting potential OO adoption in the US, Fichman & Kemerer (1993) developed a framework to analyse software process adoption (see Figure 1). The vertical axis reflects Roger's Diffusion of Innovation (1995) view of organisational adoptability: relative advantage; compatibility; complexity; trialability; and observability. The horizontal axis reflects economic factors affecting adoption at the community or industry level: prior technology drag; irreversibility of investments; sponsorship; and expectations. These axes combine to form four quadrants, implying distinctive adoption trajectories. Empirical testing of the grid on OO adoption was conducted in 1997 using case research methodology (Fichman & Kemerer 1997).

Organisational adoptability	Low	High
High ?	**Niche** – Adoption will start out fast among adopters who are relatively insensitive to standards issues or who have optimistic expectations about future levels of adoption.	**Dominant technology** – The technology will be rapidly adopted as a dominant process technology. It will face relatively low barriers to individual or community adoption.
? Low	**Experimental** – The technology will need to evolve before it is widely adopted by mainstream organisations as a dominant technology	**Slow mover** – The technology will diffuse steadily but slowly because of the difficulty individual or community adoption.

Low ? *Community adoptability* ? High

Figure 1: Software Process Technologies Adoption Grid (Adapted from Fichman & Kemerer 1993, p.10)

In the US, Fichman and Kemerer (1993, 1997) found OO methodologies to be at the *experimental stage* of adoption and hence, will need to evolve before they are widely adopted as a dominant technology by mainstream organisations. OO was also predicted to have low acceptance in large in-house IS organisations.

In Australia, adoption of the OO approach appears to be growing, judging from the number of commercial (non-academic) OO conferences run every year and the steady increase in empirical studies of OO software practice between the years 1985 to 1995 (Reed 1995). Reed also noted that a number of consultancies and development houses specialising in the use of OO techniques have increased, and an Australian commercial object-oriented language 'Ochre' has been produced.

Applying Fichman & Kemerer's (1993) grid, Dick & Rouse (1994) studied OO adoption in the Australian context. They conducted an exploratory study of four organisations in Sydney and Melbourne and found that OO in Australia has reached the *niche stage* – adoption will start out fast among adopters who are relatively insensitive to standards issues and/or those who have optimistic expectations about future levels of adoption.

RESEARCH ISSUE AND METHOD

The topic of OO adoption can be divided into three sub-topics corresponding to three phases in the overall OO adoption process: *pre-adoption*, *adoption* and *post-implementation*. The research questions addressed by this study are:

- Why does a large Australian organisation choose to adopt OO?
- What is the process used by a large Australian organisation in adopting OO?

- How does a large Australian organisation evaluate OO adoption?

Case Study

A large Queensland government organisation was chosen as the initial case study for this research project. A project team consisting of four personnel and two consultants had just completed an OO project (referred to as 'the project') at the time data collection was conducted. The project commenced in early March 1999 and was completed in late October 1999. The project's timeline is presented in Table 2.

Adoption Phase	Duration	Description
Pre-adoption phase	2.5 months	Project planning began with the evaluation of benefits and risks. Request for Proposals from leading IT consultancies were distributed. At the end of this phase, the go-ahead was given by the steering committee.
Adoption	6 months	Project commenced and completed with post-project presentations to the Steering Committee and internal staff.
Post-implementation	2 weeks	Post-implementation evaluation of the project.

Table 2: Project Timeline

In-depth interviews were conducted based on the research questions presented earlier. Informants provided data on OO adoption in the recent OO project that they were involved in. Informants were selected on the basis that they had input into the decision to adopt OO, experience in assisting the organisation in making the decision to adopt OO, input and/or experience in assisting the transition process from traditional to OO methods, input and/or experience in the management of OO projects and, input and/or experience in evaluating OO projects. The informants were the IT Director, the Project Manager and one Analyst/Programmer. The use of multiple informants with differing job tasks and responsibilities provided information on the different phases of the OO adoption process, and provided information from different stakeholder perspectives.

FINDINGS FROM THE CASE STUDY

In this section, the findings of each of the three research questions are addressed in turn. Firstly, the motivation of a large Australian organisation to adopt OO is discussed, then the adoption process is described, followed by the procedure to evaluate the adoption project.

Why Does a Large Australian Organisation Choose to Adopt OO?

In the organisation, OO technologies were selected as a consequence of business and technology analysis. The project was regarded as an initial step towards achieving the organisational goals of encouraging reuse in system development, discouraging system development from scratch, and proactivity towards technological needs. The decision was affected by the evaluation of a number of factors. These factors can be classified according to importance. From the in-depth interviews, the factors affecting the decision to adopt OO were ascertained (summarised in Table 3).

Factors of Primary Importance	Factors of Secondary Importance
The technology chosen must • comply with open, standards-based solutions and not be locked in to particular vendor(s); • be consistent with existing values and organisational goals/needs; • produce visible results at the end of a project; • show promise as a pervasive technology; and • allow for experimentation on a limited basis.	• Existence of a large, mature base of existing adopters; • investments in the technology (including staff education, staff training and other financial investments); and • the need for a champion sponsor to define, set standards, financially support and promote the adoption of the technology.

Table 3: Factors affecting the decision to adopt OO.

Perceived risks. In assessing the perceived risks of adoption, the following were considered serious risks, in order of seriousness – from most serious to least serious:

1. Requirement for extensive training and education;
2. Inability to enforce reuse;
3. Unavailability of stable implementation tools (e.g. programming languages);
4. Incompatibility with existing systems;
5. Lack of suitable methodology to support system development;
6. Requirement of high investment; and
7. Requirement for organisational restructuring.

Contrary to the literature relating to remuneration of OO development staff (Federowitz & Villeneuve 1999; Classe 1995; Taylor 1992), the risk of a change in reward and remuneration of employees was deemed not applicable to the project. The chosen organisation was a government organisation, and as such, remuneration based on performance was not practised.

Mitigation strategies. Of the large number of mitigation strategies suggested by the literature (Federowitz & Villeneuve 1999; Heist & Allen 1999; Hantos & Joseph 1997; Fichman & 1992), the organization used two main strategies to reduce risks of adopting OO:

- A small scale, pilot project to test out the adoption process and reveal the issues relating to OO adoption;
- The use of external consultants to aid in education and training of internal staff to sharpen their skills and reduce the impact of a paradigm shift in mindset.

Making the decision. A steering committee made the decision to go ahead with the project. At this stage, a go-ahead for OO technologies was given, but specific requirements and proprietary products were not selected yet. This steering committee consisted of members at various levels of the organisation and backgrounds, including both business and technical personnel. The steering committee formalised the decision and provided a formal organisational structure for supporting the project.

This part of the research question identified the process by which a large Australian organisation would evaluate the benefits and risks of OO adoption, to arrive at a decision to adopt. At the organisation, the following benefits were sought:

- greater resilience to changes in the business;
- increased reuse in system development;
- higher quality systems; and
- reduction of system development time.

Having identified the benefits, costs and risks, the organisation made the decision to proceed with the adoption.

The OO Adoption Process Used by a Large Australian Organisation

Once the decision has been made to adopt OO, the second phase in the adoption process starts: adoption of OO methods by the development team in an IS project. Few studies have focused on problems associated with adopting OO in a systems development project and hence, little is known about the strategies that Australian organisations use to address the process of OO adoption. The two main schools of thought concerning the process of OO adoption are the revolutionary and the evolutionary approach.

Revolutionaries, such as Booch (1989) and, Coad and Yourdon (1991) believe that 'object-orientation is a radical change that renders conventional methodologies and ways of thinking about design obsolete' (Fichman & Kemerer 1992, p.22). Under the revolutionary approach, OO techniques take the place of traditional methodologies in a 'plunge' manner (Hardgrave

1997). The *revolutionaries* feel that the separation of data and processes as distinct entities is outdated and that these methodologies should be 'thrown out' (Fichman & Kemerer 1992; Garceau et al. 1993). Also known as 'complete rebuild' by Taylor (1992, p.293), it is advocated simply to 'pull out all the existing systems and rebuild them using OO SDMs.' In this school of thought, the traditional methodologies should not be retained when adopting OO.

Synthesists or evolutionists see object-orientation as simply an accumulation of sound software development principles that adopters can graft onto their existing methodologies with relative ease (Fichman & Kemerer 1992). Under the evolutionary approach, traditional methodologies are integrated with OO concepts to facilitate the adoption process (Hardgrave 1997). The *evolutionists* or *synthesists* (Coad & Yourdon 1991) attempt to integrate OO with the traditional methodologies.

Several authors (Fichman & Kemerer 1992; Taylor 1992) assert that the evolutionary approach has gained greater acceptance and has been more productive. Also known as 'graceful migration', Taylor (1992) asserts that this is the only realistic approach. There are several reasons for this:

- *Acceptance to change*. The introduction of OO in a gradual process increases the development team's acceptance to change in studying the new methodology;
- *Resources for retraining*. Off-the-job training is reduced as the development team trains on-the-job by integrating the traditional tools with OO tools; and
- *Comparison of tools and techniques*. The on-the-job training facilitates increased understanding of the tools besides providing the team with a comparison of the effectiveness of the different tools in different situations.

In the case study, the first and main issue encountered in the project was the difficulty of managing the project. The project manager had to be technically and managerially competent to manage the first OO project in the organisation. As a solution, two project managers were appointed: a project manager within the organisation (managerial); and a consultant project manager (technical). This provided the technical and business mix needed to achieve the project goals.

Second, there was a need for extensive education and training for staff working on the project. External consultants who were knowledgeable with the technologies were employed to provide JIT-training and JIT-education. Contrary to literature associated with the process OO adoption (Classe 1995; Hantos & Joseph 1997), there was little resistance to change and few problems with shifts in system development mindsets. This was due to the personality traits of project personnel, an enthusiastic attitude towards a small, well-managed project, and the possibilities of career progression based on skills developed from the project.

The third issue concerned the selection of specific technologies for the middleware architecture. The consultants recommended appropriate technologies to be included. These technologies were compliant with open, standards-based implementations, as specified by the organisation.

The fourth issue concerned the testing of the systems. Component-, interface- and system integration testing were performed. However, two out of three systems that were being tested with the middleware architecture were testing systems, not production systems. This reduced the risks of disruptions to existing systems. This was a risk mitigation strategy that was used in the process of adoption.

The fifth was the issue of documentation. It was found that even though documentation standards exist in the organization, they were not rigorously adhered to in the development of systems. In the project, documentation was a problem, because the OO documentation style did not sit well with the organisations' documentation standards. This issue was not resolved.

Adoption Style. A proactive evolutionary adoption style was used by the organisation in adopting OO. The technologies chosen grafted onto and/or improved the organisation's existing practices and designs. There are two components to this adoption style: 'proactive' and 'evolutionary'.

A proactive style ensured that the organisation would always attempt to stay ahead of its competition. 'Staying ahead' does not necessarily mean adopting superior technologies. At the organisation, 'staying ahead' is achieved by understanding how OO can contribute to the organisation's goals.

An evolutionary style ensured that the path of least resistance be taken when choosing technologies. With evolutionary style, the impact on the skills and experience of existing staff would not be revolutionary. An evolutionary mindset ensured that a change in skills be perceived as an enhancement to the individual's existing set of skills. Also, legacy systems are perceived to be enhanced, not replaced by new technologies.

How Does a Large Australian Organisation Evaluate OO Adoption?

As soon as a software product developed using OO methods is implemented, organisations are interested in evaluating the success (or not) of the project and of OO adoption at large. Little evidence exists about the actual benefits or productivity improvements that have been realised by companies that have adopted OO. This could be due to the fact that the measurement of productivity enhancements in OO projects is significantly different from non-OO projects (Caspers 1994; Barnard 1998). OO attempts to increase productivity in all phases of software development (Caspers 1994). It also attempts to facilitate the creation of highly-reusable and high quality software in the development of future systems. Therefore, the timeline for measuring project success is markedly different than in traditional systems (Fichman & Kemerer 1993).

Although success of OO developed systems can be measured in a newly developed OO system, a more useful measurement of OO success is the reusability of its objects in future OO systems (Barnard 1998). The assumption is made that the objects developed in the current system have been tested rigorously to reduce the likelihood of errors and hence, are of high quality. These generic objects can then be utilised in future projects. The use of these generic and high quality objects is expected to increase the quality of future projects and, at the same time, reduce development time (Barnard 1998).

One way to measure project success is through the use of software metrics. However, standard metrics used with traditional methodologies are limited in their ability to describe true OO analysis, design and code (Barnard 1998). Because OO success spans multiple projects, the issue of timeline and linkages between projects become a crucial issue. Measurements must be taken across projects, by determining the reusability of objects from one project to another. Chidamber & Kemerer (1994) proposed a suite of metrics with measurements such as: *depth of inheritance tree* to represent potential reuse of classes in a system, *number of children* to be an indirect measure of reuse capability and testing effort; and *lack of cohesion in methods* to measure the quality of system design. Alternatively, a function-point-like measure for OO software was developed by Antoniol, Lokan, Caldiera & Fiutem (1999).

The case study project had success measured *during* the adoption process and *at the end* of the project. During development, a standard project management software package was used to monitor and control the project. At the end of the project, a post-implementation review was performed to evaluate the success of the project. The post-implementation review defined the parameters by which success is measured.

Three major measures of success for the project were: the project's completion within its allocated budget; the project's completion within its timing schedule; and the effectiveness of the project output. Effectiveness in the project outcome was defined as the achievement of the objectives it had set out to achieve, the achievement of its implementation objectives, and the achievement of all of the key criteria laid down for a middleware architecture. These key criteria were security, scalability and reliability, reusability and extendibility, auditability, and performance. Based on a subjective evaluation by the project managers, the project was deemed to have met all the three objectives for effective project output.

No formal metrics (traditional or OO) were used to evaluate the success of the project. This was possible and acceptable for the project because it was a proof-of-technology pilot. This method of measurement may not be ideal for other OO projects. Project managers should look into OO software metrics, as suggested by Harrison, Counsell & Nithi (1998), Berard (1996) Chidamber, Darcy & Kemerer (1996) and Caspers (1994) as operational measures for OO project success. The suite of OO design metrics proposed by Chidamber, Darcy & Kemerer (1996) have become the *de facto* standard in the US, against which new proposed sets of metrics have been judged, compared, and evaluated.

ISSUES FOR FURTHER RESEARCH

Insights into the adoption process for organisations that have successfully or unsuccessfully adopted OO can provide practitioners with information about the important/crucial factors that could make OO a viable investment. Since the area of OO adoption and transition has not been well researched in Australia, there is still much scope for contributing to knowledge by focusing a research project on the adoption of OO by Australian organisations.

Modifying the research parameters. The same research could be replicated in other government and private sector organisations, with the findings compared with those of the current study. Also, cross-country research on OO adoption might provide a useful basis for comparing different adoption processes across national boundaries.

Implementing a multiple case studies design. A multiple case studies design could be implemented using two categories of adopters: adopters and non-adopters and even successful versus non-successful adopters. This would increase the validity and reliability of the study on OO adoption. It would also enable tests for literal replication and theoretical replication.

Implementing a quantitative research design. This study confirmed the factors that influence adoption, the risks associated with adoption, the mitigation strategies used and the perceived benefits of OO. This study also revealed a number of additional factors and issues. These factors/issues could be operationalised in a quantitative study, using surveys, to increase external validity. This could lead to statistical generalisation of OO adoption in Australia.

Testing for managerial influence on OO project success. Most measures of OO success are directly related to the efficiencies of software design. This study demonstrated that managerial influence and support could have an impact on OO adoption success. It would be beneficial to evaluate how much of this "success" is dependent upon or attributed to the effectiveness and abilities of the project manager and others involved throughout adoption.

CONCLUDING COMMENTS

This paper has investigated the adoption of OO methods and techniques, specifically identified three phases in the adoption process, outlined research questions to study each phase, and presented findings based on a case study of a large Australian government organisation adopting OO. The output of this research is expected to add to the existing knowledge in the adoption literature. Although models of innovation adoption have been developed in the past, e.g. Rogers' Diffusion of Innovations model (1995), none of them relate

specifically to OO adoption in Australia. Therefore, this study has contributed to knowledge about OO adoption within the Australian context.

Until now, there has been insufficient research conducted to provide guidelines to practitioners in the OO adoption process. Although Fichman and Kemerer (1997) have identified lessons from early OO adopters in the United States, generalisability of such US-based findings to the Australian context has been questioned (Dick & Rouse 1994). One aim of the current study is to generate a set of guidelines that could assist Australian organisations in the adoption of OO. The outcome of this research would provide a foundation for further research using multiple case studies and quantitative techniques. The research output could facilitate more informed decision-making by organisations that are pondering OO investment. The research outcome could even serve to ease the transition process from developing systems using the traditional methodologies to the utilisation of OO techniques.

REFERENCES

Adhikari, R. (1996) Object-Oriented Development - Method To The Madness - Standard could make app building easier, *InformationWeek*, 576, pp. 1-2.

Antoniol, G., Lokan, C., Caldiera, G. & Flutem, R. (1999), A Function Point-Like Measure for Object-Oriented Software, *Empirical Software Engineering*, vol. 4, pp.263-87.

Attewell, P. (1992) Technology Diffusion and Organizational Learning: The case of Business Computing, *Organization Science*, vol. 3, 1, pp. 1-19.

Barnard, J. (1998) A new reusability metric for object-oriented software, *Software Quality Journal*, vol. 7, 1, pp. 35-50.

Berard, E.V. (1996) *Metrics for Object-Oriented Software Engineering* [Online], Available: http://www.toa.com/pub/html/moose.html, [Accessed 99].

Booch, G. (1989) What Is and What Isn't Object-Oriented Design?, *American Progammer*, vol. 2,7-8, pp. 14-21.

Brown, D. (1997) *An Introduction to Object-Oriented Analysis*, John Wiley & Sons, New York.

Caspers, J. (1994) *Object-Oriented Programming Analysis, Design and Implementation Methods*, Computer Technology Research Corp., South Carolina.

Chidamber, S. R., Darcy, P. D. & Kemerer, C. F. (1996), Managerial use of object-oriented software metrics, Working Paper Series No. 750.

Chidamber, S. R. & Kemerer, C. F. (1994), A metrics suite for object oriented design, *IEEE Transactions on Software Engineering*, vol. 26, 6, pp.467-93.

Classe, A. (1995) An object lesson, *Computer Weekly*, pp. 34-5.

Coad, P. and Yourdon, E. (1991) *Object-Oriented Analysis*, 2nd edn, Yourdon Press, Englewood Cliffs, New Jersey, USA.

Dick, M. and Rouse, A. (1994) Introducing object-oriented systems development processes: The experience of four Australian organisations, *The Information Systems Working Paper Series: Monash University*, pp. 1-8.

Federowitz, J.and Villeneuve, A.O. (1999) Surveying object technology usage and benefits: A test of conventional wisdom, *Information and Management*, vol.35,iss.6, Jun,pp.331-44.

Fichman, R.G. and Kemerer, C.F. (1992) Object-Oriented and Conventional Analysis and Design Methodologies, *Computer*, pp.22-39.

Fichman, R.G. and Kemerer, C.F. (1993) Adoption of Software Engineering Process Innovations: The Case of Object Orientation, *Sloan Management Review*, vol.34, 2, pp.7-22.

Fichman, R.G. and Kemerer, C.F. (1997) Object Technology and Reuse: Lessons from Early Adopters, *Computer*, pp. 47-59.

Garceau, L.R., Jancura, E.G. and Kneiss, J. (1993), Object-oriented analysis and design: A new approach to systems development, *Journal of Systems Management*, vol. 44, 1, pp. 25-32.

Hantos, P. and Joseph, S. (1997) Object Technology Adoption -- A Risk Management Perspective, Xerox Corporation, USA.

Hardgrave, B.C. (1997) Adopting object-oriented technology: evolution or revolution?, *The Journal of Systems and Software*, vol. 37, 1, pp. 19-25.

Harrison, R., Counsell. S. J. & Nithi, R. V. (1998), An investigation into the Applicability and Validity of Object-Oriented Design Metrics, *Empirical Software Engineering*, 3, pp.255-73.

Heist, A. and Allen, P. (1999) Australian vendor moves to object-oriented technology, *Operations Management*, vol.5, 38, 20 Sep, p.11.

Henry, S.M. and Humphrey, M. (1993) O-O vs. procedural programming languages: Effectiveness in program maintenance, *Journal of Object-Oriented Programming*, vol. 6, 3, pp. 41-50.

Johnson, R. & Hardgrave, B. C. (1999a) Object-Oriented Systems Development: Current Practices and Attitudes in Industry, *Journal of Systems & Software*, vol.48, 1, pp.5-12.

Johnson, R. & Hardgrave, B. C. (1999b) Usefulness, Social Influence, and Ease of Use of Object-Oriented Systems Development: Beliefs of Experienced OO Developers, *Journal of Computer Information Systems*, vol.39, 4, Summer, pp.26-32.

Meyer, B. (1987) Reusability: The Case for Object-Oriented Design,*IEEE Software*,pp.50-64.

Reed, K, (1995) *Software Engineering in Australia...A Brief Overview* [Online], Available: http://insect.sd.monash.edu.au/sercc/karl.html, [Accessed 98].

Rogers, E.M. (1995) *Diffusion of Innovations*, 4th edn, Free Press, New York.

Taft, D.K. (1995) OOP, there it is: Remixing, reusing existing materials, *Computer Reseller News*, 647, p. 85.

Sultan, F. & Chan, L. (2000) The Adoption of New Technology: The Case of Object-Oriented Computing in Software Companies, *IEEE Transactions on Engineering Management*, vol.47, 1,pp.106-126.

Taylor, D.A. (1992) *Object-Oriented Information Systems: Planning and Implementation*, John-Wiley and Sons, New York, USA.

COPYRIGHT