

The Effect of Reusability on Perceived Competitive Performance of Australian Software Firms

S.L.D. Ah-Fock

Department of Information Systems
University of Southern Queensland
Toowoomba 4350 QLD Australia
Email: ahfocks@usq.edu.au

A.L.M. Cavaye

Australian Graduate School of Entrepreneurship
Swinburne University of Technology
Hawthorn 3122 VIC Australia
Email: aCavaye@swin.edu.au

This study set out to investigate how software firms can manage their software development efforts in order to compete effectively. Based on previous studies, a research model was developed. Reusability was expected to positively influence process flexibility, process productivity and process predictability. In turn these software process performance dimensions were expected to influence perceived competitive performance, assessed in terms of market responsiveness and product cost efficiency. Three hundred and twenty organisations were surveyed. In the data reduction stage, the research model was revised and process efficiency, techno-regulatory flexibility, process effectiveness and labour flexibility resulted as the new software process performance variables. The revised model was tested and findings indicate that there is a relationship between reusability and techno-regulatory flexibility and market responsiveness.

Keywords: Reusability, process flexibility, process productivity, process predictability, perceived competitive performance

1. INTRODUCTION

In today's dynamic and uncertain environment, the issue of competitive performance is becoming increasingly important for software firms. Due to intense competition, software firms are pressured to respond quickly to market changes through reduced time-to-market, cost-effective and high quality products (Cusumano, 1991; Nidumolu and Knotts, 1998). *Consequently, an important question to investigate is: how can software firms improve their software development efforts in order to sustain competitiveness?*

In practice there are a number of ways in which software firms can improve their competitive performance: introduce new technologies (Porter, 1985; McNurlin and Sprague, 1998), use of the spiral model for project management (Boehm, 1988) and adopt object-oriented methodologies

Copyright© 2003, Australian Computer Society Inc. General permission to republish, but not for profit, all or part of this material is granted, provided that the JRPIT copyright notice is given and that reference is made to the publication, to its date of issue, and to the fact that reprinting privileges were granted by permission of the Australian Computer Society Inc.

Manuscript received: 4 March 2000
Communicating Editor: Graham Low

(Yourdon, 1992). While each of these is important, this study focuses on yet another way to improve competitive performance: software process management.

Software process improvement has received considerable attention in both industry and academia (Humphrey, 1989; Paulk *et al*, 1993; El Eman *et al*, 1998). One of the main objectives of process-based approaches to software development is reusability of software components. Reusability can affect the competitive performance of software firms through its effect on three software process performance dimensions: process flexibility, process productivity, and process predictability (Cusumano, 1991; Joos, 1994; Lim, 1994). Reusability is an important construct but only a few studies (most recently Nidumolu and Knotts, 1998) have attempted to empirically investigate its effect on software process performance dimensions and competitive performance of software firms. This study set out to contribute to the reusability debate.

This paper first discusses findings from published literature about reusability in the software development process. Then, the research model is presented and the method used to collect the data to test the research model is outlined. Next, findings from a survey of Australian software firms are presented and discussed. Finally, the conclusion outlines implications for practitioners and suggests avenues for further research.

2. REUSABILITY IN THE SOFTWARE DEVELOPMENT PROCESS

This section provides an overview of the published literature to highlight current findings about reusability, the software development process performance dimensions, and competitive performance. Each of the constructs and relationships among them are discussed.

2.1 Perceived Competitive Performance

Competitive performance is a broad and multidimensional concept that is difficult to define and measure (Dess and Robinson, 1984). This study focused on “perceived” competitive performance, i.e. on the way in which an organisation perceives itself with regards to its competitors in terms of competitive performance dimensions (such as product quality and efficiency). Thus, in this study, competitive performance was defined as “... how a software firm views itself relative to its competitors in terms of market responsiveness and product cost efficiency” (Nidumolu and Knotts, 1998).

The two dimensions *market responsiveness* and *product cost efficiency* are particularly relevant for software firms because these dimensions illustrate key advantages that software firms can achieve by improving their internal process to develop software (Boynton *et al*, 1993; Nidumolu and Knotts, 1998). Market responsiveness refers to the level of timeliness with which a software organisation can respond to market changes when compared to its competitors (Olsen, 1995); product cost efficiency describes the efficiency with which a software organisation can produce software from a cost perspective when compared to its competitors (Nemetz and Fry, 1988).

2.2 Reusability In Process-Based Software Development

Improvements in the software development process can enhance the competitive performance of software organisations. Prior research in software process engineering (Cusumano, 1991; Paulk *et al*, 1995; Deephouse *et al*, 1995) has identified a number of factors associated with process improvement, e.g. reusability, customisability, skills standardisation, and project management and planning. While each of these contribute to perceived competitive performance, reusability is often cited as a *key* component of process-based improvement of software development (Cusumano, 1991; Paulk *et al*, 1993; Nidumolu and Knotts, 1998). According to Mili *et al* (1995), experts perceive reusability as the “... only realistic solution to problems in software development”.

Reusability is the recycling of outputs from one software development project to another project (Nidumolu and Knotts, 1998). Reusability is believed to increase process performance by increasing flexibility, productivity, predictability and quality (Bollinger and Pfleeger, 1990; Basili and Rombach, 1991; Karlsoon, 1995; Thorne, 1998; Yongbeom and Stohr, 1998). In turn, these process performance dimensions are believed to increase competitive performance (Cusumano, 1991; Nidumolu and Knotts, 1998). Yongbeom and Stohr (1998) emphasise the causal relationship between reusability and the software process performance dimensions: “Because fewer total lines of code need to be written, reuse can increase productivity” and “Since reusable software resources should be rigorously tested and verified, reuse has the potential to increase quality, reliability, maintainability and predictability”.

An often quoted example of an organisation that used reusability with great success is the Raytheon Company. A software reuse program helped this company increase productivity by 60 per cent and also allow Raytheon improve product quality and process flexibility (Haley, 1996). Other published cases (including Digital Equipment Corporation and BTG) also show the benefits resulting from a software reuse program (Incorvia *et al*, 1990).

However, while research has outlined the benefits of software reuse and has documented some successful cases, its promise has been largely unfulfilled (Biggerstaff and Ritcher, 1987). Some of the reasons for the failures of reuse programs include: costs, lack of management commitment, and difficulties in measuring reuse (Verschoor and Low, 1993; Thorne, 1998; Yongbeom and Stohr, 1998). Also, while it is commonly claimed that reusability increases competitive performance through its effect on software process performance dimensions, little empirical research has been carried out to test these relationships (Nidumolu and Knotts, 1998). This study set out to address this gap in the research about reusability.

2.3 Software Process Performance Dimensions

There are three main software process performance dimensions that were considered: process flexibility, process productivity, and process predicability.

Process flexibility. Process flexibility is defined as the “... speed with which an organisation’s software development approach can respond effectively to changes in the organisation’s environment” (Nidumolu and Knotts, 1998). While flexibility is a key factor in increasing market responsiveness, traditionally, flexibility was believed to negatively impact on cost-efficiencies. However, recent research (Kotha, 1995; Jordan and Graves, 1995) have shown while firms need to invest to have a flexible automated production line, they need not compromise cost efficiencies. Hence, increases in process flexibility are likely to positively influence both market responsiveness and product cost efficiency.

Process productivity. Productivity is defined in its simplest form as the ratio of outputs to inputs (Banker and Kauffman, 1991). According to Scudder and Kucic (1991), both efficiency and effectiveness must be taken into consideration when measuring productivity. Efficiency is concerned with the resources consumed in producing a given application in a timely manner, while effectiveness is concerned with the quality of finished products and its appropriateness to solve the initial problem. Both efficiency and effectiveness have been found to affect competitive performance. For example, Cusumano (1991) found that increases in process productivity (efficiency and effectiveness) in software organisations in Japan helped the Japanese firms increase their global market share. In a similar anecdote, increases in process productivity at GTE Data Services helped the organisation save \$10 million (Prieto-Diaz, 1991). Thus both dimensions of productivity are likely to positively influence market responsiveness and product cost efficiency.

Process predictability. Process predictability is defined as “... the ability of the software firm to accurately estimate the needed resources, time, performance, quality and functionality of its software projects” (Dowson, 1993). According to Boynton and Victor (1991), a predictable process is at the basis of software firms’ systematic continuous improvements that allow them to respond quicker to the market. At the same time, predictable processes can provide cost advantages through the process of “dynamic stability” (Boynton *et al*, 1993). Thus, process predictability is likely to positively influence both market responsiveness and product cost efficiency.

2.4 Research Framework

The question that this study sought to answer was:

To what extent does reusability influence perceived competitive performance of software firms through its effect on process flexibility, process productivity and process predictability?

Previous research suggests that positive relationships can be expected between reusability and software performance dimensions and also between software performance dimensions and competitive performance. As outlined earlier, reusability is expected to positively influence software process flexibility (Kotha, 1995; Nidumolu and Knotts, 1998), software process productivity (Cusumano, 1991; Nidumolu and Knotts, 1998) and software process predictability (Boynton and Victor, 1991; Boynton *et al*, 1993). In turn, the software process performance dimensions are expected to influence perceived competitive performance, both market responsiveness (Olsen, 1995) and product cost efficiency (Nemetz and Fry, 1988). These relationships are summarised in the initial research model presented in Figure 1.

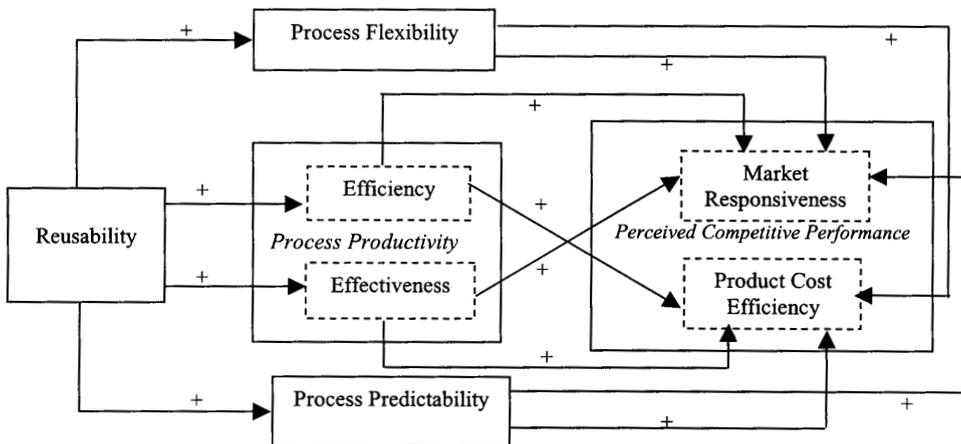


Figure 1: Initial research model

3. SURVEY OF AUSTRALIAN SOFTWARE FIRMS

In order to test the relationships in the initial research model, a survey of Australian software firms was conducted. A questionnaire was designed and mailed to 320 commercial software organisations in Australia.

3.1 The questionnaire

The questionnaire consisted of seven sections. The first section consisted of a number of questions about the demographics of the organisations. The next five sections asked questions pertaining to

the constructs under investigation: reusability, process flexibility, process productivity, process predictability, and perceived competitive performance. The questions in the final section were concerned with mechanisms to promote software reuse; these questions were included to see to what extent organisations were using each of the five mechanisms presented in the questionnaire to promote reuse. All the constructs were measured using a five point Likert scale. The questionnaire was developed using measurement items from previously developed and validated research instruments. The independent variable “reusability” was measured with items developed by Apte *et al* (1990) and Nidumolu and Knotts (1998). Five reusable components were considered, including requirement specifications, design specifications, software code, test data and documentation.

The software process performance construct was measured using the three dimensions “process flexibility”, “process productivity”, and “process predictability”. Process flexibility was measured using five dimensions developed by Bantel (1993): labour-supply, customer needs, competition, regulation, and technology. Bantel (1993) identified five separate dimensions; Nidumolu and Knotts (1998) found that these dimensions are not distinct. This study adopted the findings of Nidumolu and Knotts (1998). Hence, process flexibility was considered as one entity in this study and the five dimensions of process flexibility were used as five measures of process flexibility.

Process productivity was investigated by including questions on both quantitative and qualitative measures for software process productivity. Quantitative items included lines of code, time and cost, and function point analysis; qualitative items included the practice of structured design and project control. Individual items measuring productivity were adapted from the following studies (Lane, 1998; Deephouse *et al*, 1995; Henderson and Lee, 1992).

Process predictability is closely related to process productivity. For example, improvements in estimating plans and resources lead to shorter schedules and hence increase productivity (Paulk *et al*, 1995). Thus, process predictability can be viewed in terms of “efficiency” as well as “effectiveness”. However, these two dimensions of process predictability have not been previously empirically tested. Hence, process predictability was considered as one entity in this study. Individual items measuring process predictability were adapted from Nidumolu and Knotts (1998) and Boynton *et al* (1993).

The dependent variable “perceived competitive performance” was measured by questions about *market responsiveness* and *product cost efficiency*. Individual items for these measures were derived from Nidumolu and Knotts (1998).

3.2 Details of the survey

The survey was mailed to a total of 320 commercial software organisations in Australia. Specifically, organisations were selected out of the 728 organisations listed by the “Australian-on-Disc” database as being involved in commercial software development in the states of Queensland and New South Wales. In order to obtain a representative sample and to minimise sampling error, the organisations were randomly selected.

A total of 118 questionnaires were returned. Of these 118 questionnaires, 72 were unusable responses: 23 were incomplete responses (returned by software firms who identified themselves as non-software developers and/or non-reuse practitioners); 49 questionnaires were returned as undeliverable (due to inaccurate addresses on the database). Hence, only 46 out of the initial 320 mailed questionnaires were valid responses.

The 46 valid surveys received reflect a valid response rate of 14.4%. The response rate (14.4%) is considered poor. The high percentage of undeliverable questionnaires (15.3%) clearly contributed to the poor result. However, there is no publicly accessible database which specifically identifies

Australian commercial software organisations; the researchers had no choice but to use a more general database for this survey.

As suggested by Armstrong and Overton (1977), early and late respondents were compared for late-response bias. Based on the ratings of competitive performance and reusability no significant differences were found.

4. SURVEY FINDINGS

This section provides a summary of the major findings of this study. First, descriptive information about the 46 respondents is presented. Then, the reliability and validity of the independent, intermediate and dependent variables are examined. Finally, the results of simple and multiple regression analysis are discussed.

4.1 Descriptive findings

The majority (78.4%) of the 46 organisations can be considered as mature as they had been in operation for at least five years (refer Table 1). These organisations developed software for a wide range of business sectors with distribution concentrated in retail sectors (21.7%), education (17.4%) and finance and insurance (17.4%).

<i>Organisation Maturity</i>	Frequency (N=46)	Percentage
Less than 5 years	10	21.7%
5 to 10 years	15	32.8%
11 to 15 years	11	23.9%
More than 15 years	10	21.7%

Table 1: Distribution of organisation’s maturity

Most of the organisations that responded (93.5%) were small organisations with less than 10 full-time software developers and a budget/turnover of less than \$5 million (refer Table 2). These statistics are consistent with previous finding from the Australian software industry. In 1995, Madden (1995) found that “...97% of computer services businesses employ fewer than 20 people”.

<i>Size of Organisation</i>	Frequency (N=46)	Percentage
Budget/turnover		
Less than \$5 million	43	93.5%
\$5 million to \$9 million	0	0.0%
\$10 million to \$49 million	2	4.3%
Don’t Know	1	2.2%
No. of software developers		
Less than 10	43	93.5%
11 to 49	0	0.0%
50 to 99	2	4.3%
100 to 300	1	2.2%

Table 2: Budget/turnover and number of software developers

The distribution of the number of years the respondents have been practicing software reuse is shown in Table 3. As can be seen, most of the respondents (67.3%) have been practicing software reuse for at least five years. The findings also confirm that reusability has been around for at least fifteen years and has been used by Australian software organisations.

<i>Years of reuse practice</i>	Frequency (N=46)	Percentage
Less than 5 years	15	32.6%
5 to 10 years	14	30.4%
11 to 15 years	7	15.2%
More than 15 years	10	21.7%

Table 3: Distribution of the number of years of software reuse practice

A special section in the questionnaire requested the respondents to indicate what mechanisms to promote software reuse were used by their organisations. Figure 2 displays the results. Most of the respondents (63%) indicated that they have a library of reusable resources for at least some of their software projects. A large majority (54.1%) also pointed out that they consider reusability as a formal part of the software development process for at least some of their projects. The other three reuse mechanisms mentioned in the questionnaire: reward systems for creating reusable resources; reward system for reusing existing resources; and reusability as part of formal appraisal are hardly used by the respondents. This finding is understandable particularly when the size of the respondent firms is taken into account. Most of the respondents were small firms while the three remaining reuse mechanisms are costly and know to be used by large organisations (Nidumolu and Knotts, 1998).

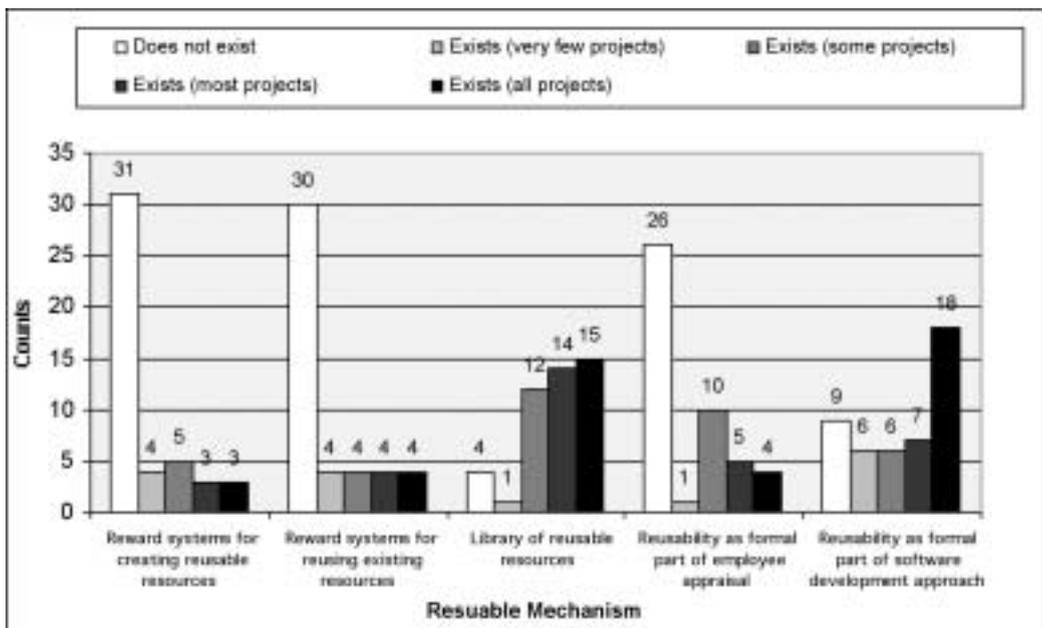


Figure 2: Proportion of organisations having each reuse mechanism

4.2 Reliability and validity tests

The reliability and validity of the independent variable (reusability), the intermediate variables (process flexibility, process productivity and process predictability) and the dependent variables (market responsiveness and product cost efficiency) were tested using reliability and factor analysis respectively.

In the data reduction stage, some of the items were removed and some of the items loaded under different variables. Process flexibility split into two factors: *techno-regulatory flexibility* and *labour flexibility*. Process productivity and process predictability did not remain distinct variables but due to their interrelatedness (as previously discussed), loaded together on two factors which were labelled *process efficiency* and *process effectiveness*. Perceived competitive performance loaded on two factors (market responsiveness and product cost efficiency) and reusability on one factor as expected. The results are presented in Table 4.

	Items Retained	Cronbach Alpha	Factor Loadings	Variance Explained
Dependent Variables				
Market Responsiveness	3	0.806	>0.58	53.8%
Product Cost Efficiency	2	0.711	>0.79	22.8%
Intermediate Variables				
Process efficiency	5	0.869	>0.58	34.2%
Techno-Regulatory Flexibility	4	0.898	>0.77	23.9%
Process effectiveness	3	0.827	>0.73	11.9%
Labour Flexibility	2	0.839	>0.86	9.1%
Independent Variable				
Reusability	5	0.814	>0.65	59.1%

Table 4: Summary of the results of reliability analysis and factor analysis

The Cronbach Alpha scores for the dependent, intermediate and independent variables were considered good, except for product cost efficiency which was very close to the acceptable level of 0.7 (Nunnally, 1978). The factor loadings for the items retained for the dependent, intermediate and independent variables were well above 0.5 level, hence they displayed adequate convergent and discriminant validity.

After the data reduction stage, the research model was revised with process efficiency, techno-regulatory flexibility, process effectiveness and labour flexibility as the new software process performance variables. The revised research model is presented in Figure 3 below.

This revised research model was tested to determine the effect of reusability on perceived competitive performance through its effect on the four (revised) software process performance dimensions.

4.3 Testing of revised research model

Simple and multiple regression analysis were used to test the revised research model. Simple regression analysis was used to test the strength of the relationships between the independent variable reusability and the process performance variables: process efficiency, techno-regulatory flexibility, process effectiveness, and labour flexibility. Multiple regression analysis was used to test the strength of the relationships between the four process performance variables and the dependent

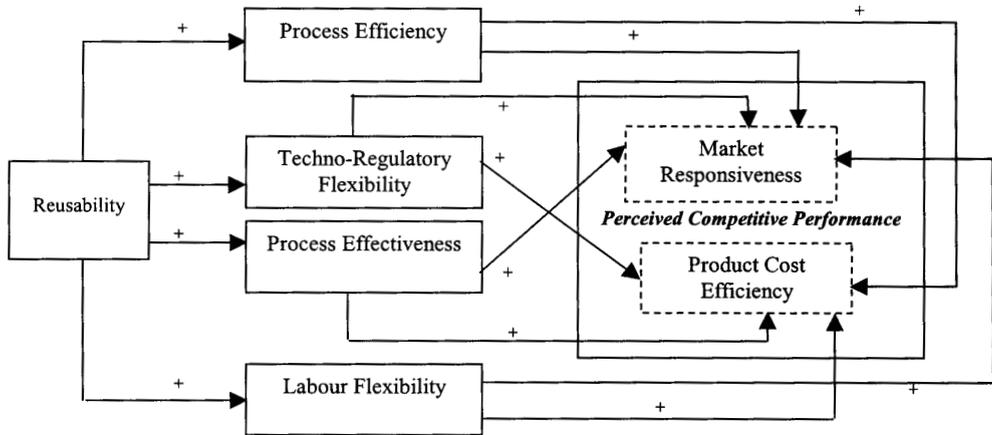


Figure 3: Revised research model

variables market responsiveness and product cost efficiency. These results are presented and discussed below.

Reusability and the software process performance variables

Table 5 shows the results of the simple regression analysis between reusability and the four software process performance variables: techno-regulatory flexibility, process efficiency, process effectiveness and labour flexibility.

Independent Variable	Dependent Variable	F-value	Adjusted R ²
Reusability	Techno-Regulatory Flexibility	0.043	0.071
	Process effectiveness	0.050	0.065
	Process Efficiency	0.837 (ns)	0.001
	Labour Flexibility	0.244 (ns)	0.009

ns = not significant

Table 5: Results of simple regression analysis

Techno-regulatory flexibility. The survey findings indicated that even though there was a significant relationship between reusability and techno-regulatory flexibility at the 5 percent level, the relationship was weak (adjusted R² = 0.071). A possible reason for the weak relationship is the low response rate. Also, a philosophy of reuse may often impose constraints on the process which reduces its flexibility (Nidumolu and Knotts, 1998).

Even though the relationship is weak, with rapid changes in laws and technologies in the software industry (both hardware and software), software firms need to be more vigilant and “techno-regulatory” flexible in order to remain competitive. This study provides further support to previous studies (Cusumano, 1991; Swanson *et al*, 1991; Reifer, 1992) that reusability helps improve process flexibility (i.e. technological and regulatory included) as it reduces development time and also makes it easier to modify the developed software. However, as only a weak relationship was found, further research is suggested in this area.

Process effectiveness. The survey findings also indicate that there was a relationship between reusability and process effectiveness. However, the adjusted R² value (0.065) showed that the

relationship was weak. The low response rate and the difficulty associated with measuring both quality and reusability is believed to have had an impact on the relationship being tested. An Australian study (Verschoor and Low, 1993) found that a large number of software organisations have difficulties measuring both reuse level and benefits received from it.

Even though only a weak relation was found, this study provides further support to previous research (Swanson *et al*, 1991; Joos, 1994; Lim, 1994) that reusability increases process effectiveness. As more components from previous projects are used in a new project, the more of its capabilities become known and predictable and hence the less the likelihood for errors (Reifer, 1992).

Process efficiency and labour flexibility. The results of the simple regression analysis indicated that reusability had no significant relationship with either process efficiency or labour flexibility. The lack of support can be explained to some extent by the size of the organisation that responded. As a large proportion (93.4%) of the organisations that responded were small software firms, it is believed that the level of reuse (even though rated high in some cases) was not significant enough to have a major impact on both process efficiency and labour flexibility.

Indeed, as most of the respondents were small organisations, labour flexibility (i.e. changes in quantity and quality of software developers) may not have a major effect on the organisation as the labour turnover is low and stable. This point is supported by the fact that most organisations (above 90%) indicated that they would at least match their competitors in terms of responding to the market should the quality and quantity decrease.

Software process performance variables and perceived competitive performance variables.

Table 6 shows the results of the multiple regression analysis between the software process performance variables and the perceived competitive performance variables market responsiveness and product cost efficiency. The results are discussed in turn.

Variables	Market Responsiveness		Product Cost Efficiency
	Beta value	t-value	t-value
Techno-Regulatory Flexibility	0.484	0.001	0.880(ns)
Process Efficiency	0.463	0.162(ns)	0.272(ns)
Process Effectiveness	0.093	0.490(ns)	0.549(ns)
Labour Flexibility	0.116	0.389(ns)	0.856(ns)
Adjusted R ²		0.222	-0.056

ns = not significant

Note: The Beta value for Product Cost Efficiency was not included because none of the variables has a significant relationship with it.

Table 6: Results of multiple regression analysis

Market responsiveness. The survey findings indicated there was a moderately significant relationship between techno-regulatory flexibility and market responsiveness at the 5 percent level (adjusted R² = 0.222). This study’s findings reinforce the view in competitive performance literature (Bantel, 1993; Boynton *et al*, 1993; Nidumolu and Knotts, 1998) that techno-regulatory flexibility is vital for rapid market responsiveness. By being more flexible to changing technologies and changing regulations, organisations have the competitive advantage to introduce new products/services before competitors and hence minimise the risks of loosing market share.

There was no support for the relationships between the other three software process performance dimensions (process efficiency, process effectiveness labour flexibility) and market responsiveness. Several reasons are attributed to this. Even though process productivity, labour flexibility and process –demand volume are prime determinants of market responsiveness, Bantel (1993) and Nidumolu (1995) point out there are factors like project size, creativity, planning formality, environmental complexity that determine performance and market responsiveness. In this study, because of the research objectives and for parsimonious reasons, not all of the variables mentioned above were included in the research model. Though appropriate measures (e.g. random sampling) were taken to minimise the effects of these confounding variables, it is believed that these confounding variables might have impacted on the model being tested.

Regarding the poor relationship between labour flexibility and market responsiveness, the nature of the Australian software industry is believed to be the major factor that might have impacted on the relationship. As most of the organisations that responded to the survey are small software firms with fewer than ten professional software developers, labour flexibility might have little relative importance on market responsiveness.

Product cost efficiency. No significant relationship between any of the software process performance dimensions (process efficiency, techno-regulatory flexibility, process effectiveness, labour flexibility) and product cost efficiency was found. The loss of two items measuring product cost efficiency in the data reduction stage and the relatively poor reliability score (Cronbach alpha = 0.711) for the items retained to measure product cost efficiency, meant that the variable may not be a good measure. This is to some extent believed to have been responsible for the lack of support for the relationship between the software process performance variables and product cost efficiency.

The poor result for the relationship can also be explained to some extent by the small size of the respondent organisations. According to Humphrey (1996) and Broadman and Johnson (1997), many small firms have difficulties quantifying the benefits they receive from reusability. Indeed, an Australian study (Verschoor and Low, 1993) found that “No organisation was able to adequately quantify the costs and benefits in reusing software”. The measurement of benefits is made more difficult and complex as there are no standard metrics to measure the performance variables (Scudder and Kucic, 1991; Jones, 1996; Nidumolu and Knotts, 1998). Consequently, it is possible different perceptions about cost efficiency and the software process performance dimensions have resulted in variations that have caused the levels of the performance variables to be insignificant to have major impact on product cost efficiency.

5. CONCLUDING COMMENTS

This study investigated the extent to which reusability influenced perceived competitive performance of software firms through its effect on software process performance. A research model was developed from previous literature to test the research question. A sample of 320 organisations was surveyed to collect the data. Factor analysis suggested that some of the factors in the initial research model should be revised. The revised relationships were analysed using simple and multiple regression analysis. The survey findings indicate that there is a relationship between *reusability*, the software process performance dimensions *techno-regulatory flexibility* and *process effectiveness*, and the perceived competitive performance variable *market responsiveness* (see Figure 4 below). The solid lines indicate the relationships that were significant in this study; the broken lines show those that were not supported.

The above findings contribute to the reusability debate. While surprisingly no relationship between reusability and perceived competitive performance could be established in the recent US

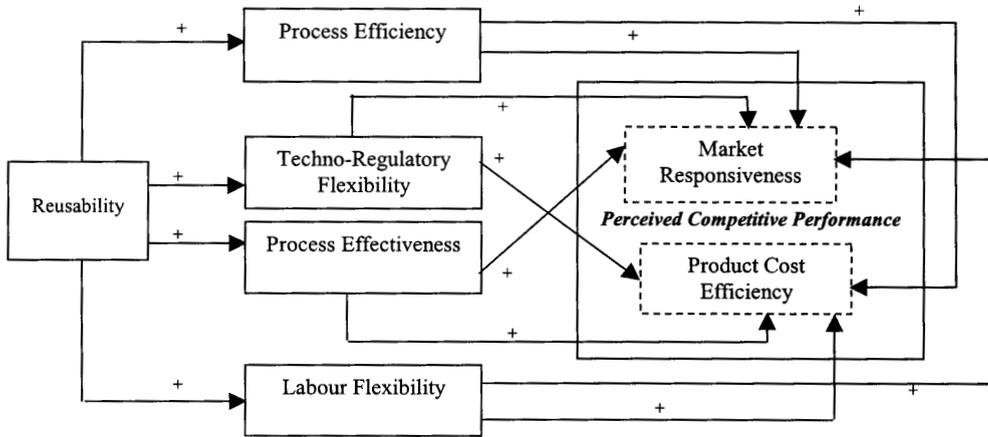


Figure 4: Final research model showing factors that influenced competitive performance

study (Nidumolu and Knotts, 1998), this study provides some support that there potentially is a relationship between reusability and perceived competitive performance. However, as the constructs under investigation are new in IS research and considering the limitations of this study, it is suggested that further empirical research be carried out in this area to further knowledge.

The findings indicate that there are weak but significant relationships between reusability and two software process performance dimensions techno-regulatory flexibility and process effectiveness. Thus, practitioners need to consider the importance of the reusability practice in improving the quality of their products as well as the importance of reusability in helping them cope with the technological and regulatory changes in the business environment.

The findings also indicated that there was a moderate positive relationship between techno-regulatory flexibility and market responsiveness. This finding is very important for management as it highlights the need for organisation to have a watching brief on technological and regulatory changes, as these changes may bring about more competition or new opportunities or both.

The lack of support for the relationships between reusability and process efficiency and those relating the software process performance dimensions and product cost efficiency highlight the necessity for management to be able to quantify the benefits and costs of the reuse program, especially if it was adopted for cost-efficiency reasons, otherwise reuse might be just a waste of time and money.

This study was limited to Australian organisations. The sample size of 320 was relatively small and the statistical power of the study is considered low with only 46 usable responses used in data analysis. The low number of respondents suggests that the results of this study must be treated with caution; the findings may not be fully representative of practices in the Australian software industry. Also, Australian software developers tend to be small organisations; hence, the findings of this Australian study are not necessarily generalisable to countries with a sizable proportion of large software development firms.

The measurement of the constructs reusability and performance is difficult. Even though appropriate measures were taken to validate the individual terms, it is likely that the interpretation of the scales may not have been uniform across respondents. Future research could be undertaken to develop more rigorous measures for these variables.

Since reusability falls in a new domain of research and the measurement instruments are relatively new, there is still much that can be done to contribute to this area of research. Future

research could further refine the measurement instruments by conducting a number of case studies. The software process performance variables and the product cost efficiency variable in particular need to be more comprehensive if they are to capture the appropriate meaning for the respective variables. Furthermore, a national study could be undertaken to generalise the results and to provide a better insight of the state of reuse in the Australian software industry. The testing of a more comprehensive model is suggested using structural equation modelling.

REFERENCES

- AMSTRONG, J. S. and OVERTON, T. S. (1977): Estimating non-response bias in mail surveys', *Journal of Marketing Research*, 14: 396–402.
- APTE, U., SANKAR, C. S., THAKUR, M. and TURNER, J. E. (1990): Reusability-based strategy for development of information systems: implementation experience of a bank, *MIS Quarterly*, 14(4): 421–433.
- BANKER, R. D. and KAUFFMAN, R. J. (1991): Reuse and productivity in integrated computer-aided software engineering, *MIS Quarterly*, 15(3): 374–399.
- BANTEL, K. A. (1993): Top, team, environment and performance effects on strategic planning formality, *Group and Organisation Management*, 18(4): 436–458.
- BASIL, V. R. & ROMBACH, H. D. (1991): Support for comprehensive reuse, *Software Engineering Journal*, 6(5), 303–319, September.
- BIGGERSTAFF, T. and RITCHER, C. (1987): Reusability framework, assessment and directions, *IEEE Software*, 4(2): 41–49.
- BOEHM, B. W. (1988): A spiral model of software development and enhancement, *Computer*, May.
- BOLLINGER, T. B. and PFLEEGER, S. L. (1990): Economics of reuse: issues and alternatives, *Information and Software Technology*, 32, December.
- BOYNTON, A. C. and VICTOR, B. (1991): Beyond flexibility: building a dynamically stable organisation, *California Management Review*, 53–66.
- BOYNTON, A. C., VICTOR, B. and PINE, B. J. (1993): Beyond flexibility: building a dynamically stable organisation, *IBM Systems Journal*, 32(1): 40–64.
- BROADMAN, J. G. and JOHNSON, D. L. (1997): *The LOGOS tailored CMMSM for small businesses, small organisations, and small projects*, LOGOS International Inc.
- CUSUMANO, M. A. (1991): *Japan software factories: a challenge to U. S. management*, Oxford University Press, New York.
- DEEPHOUSE, C., MUKHOPADHYAY, T., GOLDENSON, D. R. and KELLENER, M. I. (1995): Software process and project performance, *Journal of Management Information Systems*, 12(3), 187–206.
- DESS, G. B. and ROBINSON, R. B., Jr (1984): Measuring organisational performance in the absence of objective measures: the case of a privately-held firm and conglomerate business unit, *Strategic Management Journal*, 5: 265–273.
- DOWSON, M. (1993): Software process themes and issues, *Second International Conference on the Software Process*, Berlin, 54–62.
- EL EMAM, K. and BRIAND, L. (1997): Costs and benefits of software process improvement, International Software Engineering Research Network Technical Report, *ISERN*, 97–12.
- HALEY, T. J. (1996): Software process improvement at Raytheon, *IEEE Software*, 13(6): 33–41.
- HENDERSON, J. C. and LEE, S. (1992): Managing I/S design teams: a control theory perspective, *Management Science*, 38(6): 757–777.
- HUMPHREY, W. S. (1989): *Managing the software process*, Addison-Wesley Publishing Company, New York.
- HUMPHREY, W. S. (1996) Using a defined and measured personal software process, *IEEE Software*, May, 77–88.
- INCORVIA, A. J., DAVIS, A. M. and FAIRLEY, R. E. (1990): Case studies in software reuse, *Proceedings International Conference Software Engineering*, IEEE Computer Society Press: 301–306.
- JONES, T. C. (1996): *Applied software measurement*, 2nd edn, McGraw-Hill, New York.
- JOOS, R. (1994): Software reuse at Motorola, *IEEE Software*, September, 42–47.
- JORDAN, W. C. and GRAVES, S. C. (1995): Principles on the benefits of manufacturing process flexibility, *Management Science*, 41(4): 577–594.
- KARLSSON, E. (1995): *Software reuse: a holistic approach*, John Wiley, New York.
- KOTHA, S. (1995): Mass customisation: implementing the emerging paradigm for competitive advantage, *Strategic Management Journal*, 16(4): 21–42.
- LANE, M. (1998): *Software development productivity: an Australian study*, BIT (Hons) thesis, University of Southern Queensland.
- LIM, W. C. (1994): Effects of reuse on quality, productivity and economics, *IEEE Software*, 11(5): 23–30.
- MADDEN, R. (1995): *Computing services industry, Australia 1992-93*, Australian Bureau of Statistics, Canberra.

- McNURLIN, B. C. and SPRAGUE, R. H. (1998): *Information systems management in practice*, 4th edn, Prentice-Hall International, London.
- MILI, H., MILI, F. and MILI, A. (1995): Reusing software: issues and research directions, *IEEE Transactions on Software Engineering*, 21(6): 528–561.
- NEMETZ, P. L. and FRY, L. W. (1988): Flexible manufacturing organisations: implication for strategy formulation and organisational design, *Academy of Management Review*, 13(4): 627–638.
- NIDUMOLU, S. R. (1995): The effect of coordination and uncertainty on software project performance: residual performance risk as an intervening variable, *Information Systems Research*, 6(3): 191–219.
- NIDUMOLU, S. R. and KNOTTS, G. W. (1998): The effects of customisability and reusability on perceived process and competitive performance of software firms, *MIS Quarterly*, 22(2): 105–138.
- NUNNALLY, J. C. (1978): *Psychometric theory*, McGraw-Hill, New York.
- OLSEN, N. C. (1995): Survival of the Fastest: Improving Service Velocity, *IEEE Software*, September, 28–38.
- PAULK, M. C., CURTIS, W., CHRISSIS, M. B. and WEBER, C. V. (1993): Capability Maturity Model Version 1.1.0, *IEEE Software*, 10(4): 18–27.
- PAULK, M. C., WEBER, C., GARCIA, S. and CHRISSIS, M. B. (1995): *The capability maturity model: guidelines for improving the software process*, Addison-Wesley, Milan.
- PORTER, M. E. (1985): *Competitive advantage – creating and sustaining superior performance*, Collier Macmillan Publishers, London.
- PRIETO-DIAZ, R. (1991): Reuse in the U. S., *Proceedings of the 13th International Conference on Software Engineering*, May 13–17, Texas, USA.
- REIFER, D. J. (1992): *Software reuse for TQM*, in SCHULMEYER, G. G. and McMANUS, J. I. (eds), *Total quality for software*, Van Nostrand Reinhold, New York, 221–252.
- SCUDDER, R. A. and KUCIC, A. R. (1991): Productivity measures for information systems, *Information & Management*, 20(5): 343–354.
- SWANSON, K., McCOMB, D., SMITH, J. and McCUBBREY, D. (1991): The application software factory: applying total quality techniques to systems development, *MIS Quarterly*, 15(4): 567–579.
- THORNE, F. (1999): Cost and benefits of reuse, *The Australian Computer Journal*, 32(1), 1–8, February.
- VERSCHOOR, R. and LOW, G. (1993): Software reusability in Australia, *Proceedings of Australian software engineering Conference*, Software Engineering, 141–150.
- YONGBEOM, K. and STOHR, E. A. (1998): Software reuse: survey and directions, *Journal of Management Information Systems*, 14(4): 113–148.
- YOURDON, E. (1992): *Decline and fall of the American programmer*, Yourdon Press Englewood Cliffs, New Jersey.

BIOGRAPHICAL NOTES

Stephano L.D. Ah-Fock received a First Class Honours in Information Systems from the University of Southern Queensland (USQ) in 1999. He is currently starting on his Ph.D. He has worked as a System Analyst/Programmer for almost 5 years in Australia and Mauritius on numerous systems including ERPs. His main research interests include Software Reuse, Software Process Improvement, Component-Based Information Systems and Knowledge Management.



Stephano Ah-Fock

Professor Angele Cavaye heads the Doctor of Business Administration program at Australian Graduate School of Entrepreneurship, Swinburne University of Technology. She has a strong interest in improving support for research students and in guiding new researchers through the research process. Professor Cavaye has worked in various universities internationally and in Australia. She has taught at under-graduate and post-graduate level in both Europe and Australasia. She has published over 50 refereed research articles. Her personal research interests include e-commerce strategy and information systems issues in small business.



Angele Cavaye