Reuse: Can it Deliver Competitive Performance?

Aileen Cater-Steel
Stephano Ah-Fock

Department of Information Systems
Faculty of Business
University of Southern Queensland
Toowoomba, Queensland, Australia
Email: caterst@usq.edu.au

Abstract
This paper establishes the importance of competitive performance for software firms and highlights the potential of reusability in increasing competitive performance. The importance of reusability is discussed by examining its effect on three different software process dimensions - flexibility, productivity and predictability. From the literature, a research model is developed and used as the basis of a survey. The survey findings provide support for the relationship between reusability and some of the software process dimensions. The relationship between one of the software process dimensions and market responsiveness, a dimension of competitive performance is also supported.

Keywords

Competitive performance; reuse; process improvement; process productivity; process flexibility; process predictability; market responsiveness; product cost efficiency.

INTRODUCTION

Nowadays, software firms need to respond quickly to market demands through reduced time-to-market, cost-effective and high quality software products. This pressure on the software industry is likely to persist as there is an increasing demand for software (Yongbeom & Stohr 1998). With the increasing popularity of the Internet, competition will get even tougher as many software firms are broadening their market (BSA 1998). Hence, unless software firms ensure their competitiveness, their future could be in jeopardy (Nidumolu & Knotts 1998).

The first part of this paper discusses the perceived competitive performance of software firms and defines the dimensions of market responsiveness and product cost efficiency. Next, the popularity of reuse as a software process improvement technique is highlighted and then the software process performance dimensions: flexibility; productivity; and predictability are introduced. Based on the presented literature, a research model is presented, and the survey approach and findings are discussed. Finally, conclusions are drawn and guidelines suggested for further research.

COMPETITIVE PERFORMANCE OF SOFTWARE FIRMS

In this study, *competitive performance* is defined as how a software firm views itself relative to its competitors in terms of performance dimensions such as market responsiveness and product cost efficiency (Nidumolu & Knotts 1998). Competitive performance of software firms is important at the national, firm, and personal level. The software industry is a major contributor to the Australian economy. In the 1995-96 financial year, income from software consultancy and maintenance services, and packaged software development totalled over $3 billion Australia-wide, with 9,673 businesses employing about 55,000 persons (ABS 1997). Thus, at the national level, it is important for software firms to increase their competitive performance to ensure further national economic growth and progress (BSA 1998). At the firm level, increasing competitive performance will help software firms reduce their time-to-market and deliver higher quality products at a cheaper price, which allows above-average growth and a higher profitability (Boynton et al. 1993; Olsen 1995).

Performance is a complex and multidimensional concept. The fast-food industry identifies quality, service, cleanliness and value as major dimensions of performance whereas in the manufacturing industry, costs, flexibility and dependability are often used as measures of competitive performance. Traditionally, cost accounting data were used as the prime indicator of competitive performance (Johnson 1990). While cost is still considered an important attribute of competitive performance, nowadays there is an increasing need to consider other indicators such as flexibility, market responsiveness and product quality, due to today's complex and dynamic business environment (Johnson 1990; Nidumolu & Knotts 1998). In this study, market responsiveness and product cost efficiency were selected to represent perceived competitive performance.

Market Responsiveness

Market responsiveness relates to the speed with which an organisation responds to market changes with regard to its competitors (Nidumolu & Knotts 1998). Today, this concept is important in coping with demand uncertainties for a firm's product (Pine 1993; Olsen 1995). It is argued by Olsen (1995) that the rate at which software can be deployed to the market or customised has a more significant influence on the project decisions than quality, predictability, risk, cost or productivity. Other studies (e.g. Nemetz & Fry 1988; Boynton et al. 1993) have considered market responsiveness as a measure of competitive performance and have argued that it is indeed a very good and accurate measurement of competitive performance in today's dynamic and uncertain environment.

Product Cost Efficiency

Product cost efficiency is another dimension that is often used to measure competitive performance. Product cost efficiency of a software firm is defined in this study as its ability, relative to competitors, to produce software products from a cost perspective (Nidumolu & Knotts 1998). According to Golder and Tellis (1993), it is not only important for software firms to enter new markets rapidly but also to target mass markets rather than niche markets. A key advantage of the above approach is that firms can continue to obtain cost efficiencies that have characterised mass producers (Boynton et al. 1993). Other studies (e.g. Nemetz & Fry 1988) have considered product cost efficiency as a measure of competitive performance. They confirmed that product cost efficiency is an important indicator of competitive performance.

SOFTWARE PROCESS IMPROVEMENT

In practice, there are many ways in which software firms can improve their competitive performance. For example, new technologies can be introduced (Porter 1985), object-oriented methodologies can be adopted (Yourdon 1992), new markets can be exploited (Porter 1985; Ward & Griffiths 1998) and the spiral model can be used for project management (Boehm 1987). While each of these is important, this study focuses on yet another way to improve competitive performance: software process improvement. By managing and improving the tools, methods and practices in software development, i.e. improving the software development process, software development and consequently the performance of software organisations can be improved.

Due to its potential to increase software development process performance, process-based approaches to software development are becoming increasingly popular. Recently, a lot of attention has been given to process-based approaches both in industry and academia (e.g. Cusumano 1991; Paulk et al. 1995; Deephouse et al. 1995). Important examples of process-based approaches include the Capability Maturity Model (CMM) (Humphrey 1989), Software Process and Capability Determination standard (SPiCE - ISO/IEC 15504) (El Emam et al. 1998) and the Software Factory Approach (Cusumano 1991). In fact, Paulk et al. (1995) claim that many organisations are now realising that their fundamental problem is their inability to manage the software development process.

According to Deephouse et al. (1995), there are a number of success stories regarding software process improvement in organisations. In one example, at Schlumberger Laboratory for Computer Science, a site adopted a standard project status approach for reporting and reviews. Over a three-year period, the percentage of projects completed on schedule improved from 50 percent to 99 percent while the defect rate declined by over one-third (Wohlwend & Rosenbaum 1993). A second example reported by Humphrey et al. (1991) is on process improvements at Hughes aircraft. Over a three-year period, Hughes Aircraft achieved a $2 million reduction in budget overruns, representing six percent of budgeted work. Thus, due to the high potential of process-based approaches to improve competitive performance of software firms, it is the management of the software development process that is the focus of this study.

Prior research in software process improvement identifies a number of objectives of process-based approaches to software development including reusability, customisability, skills standardisation, product-process focus and incremental product/variety improvement (Humphrey 1989; Cusumano 1991; Swanson et al. 1991; Deephouse et al. 1995; Woods 1999). While each of these objectives is important in software process improvement, reusability is often cited as a key objective of process-based approaches to software development. According to Mili et al. (1995, p. 529), experts perceive reusability as the '… only realistic solution to problems in software development'.

Software Reuse

For a long time software development practitioners and researchers have tried to 'model' software development based on an analogy of how the manufacturing industry produces and delivers products. In the manufacturing industry, reuse of components of previous projects into new projects has helped organisations increase productivity, flexibility, predictability and has also made organisations become more able to respond to changing market demands (Humphrey 1989; Cusumano 1991). Due to the observed benefits of reusability in the manufacturing industry, reusability has become a key objective of process-based models in software development.

Reusability is defined in this study as the economies of scope that result from recycling the outputs from a software development project to another project (Nidumolu & Knotts 1998) and includes requirements and design specifications, software code, test data and documentation.

McIlroy (1968) introduced the concept of formal software reuse. His ideas grew quickly and became the basis to the `software factory' concept in the 70 's and 80's (Prieto-Diaz 1993). In

1988, Basili broadened the definition of reuse to include the use of everything associated with software project, including knowledge. According to Prieto-Diaz (1993), this revised definition has opened new doors in research in other disciplines and has contributed to the recognition that the reuse problem is ubiquitous. Today, research in software reuse encompasses a number of areas including economics, organisation performance, law, and technology. Table 1 provides a summary of key researchers of these issues.

Reuse Issue	Authors
Economic Issues	Love 1988; Apte et al. 1990; Rubin 1990; Cusumano 1991; Hooper and Chester 1991; Griss et al. 1991; Lim 1994; Due 1995
Performance Issues	Banker and Kauffman 1991; Cusumano 1991; Lim 1994; Karlsson 1995; Mayer 1997; Vendelo 1998; Yongbeom and Stohr 1998
Technology Aspects	Ramamoothy et al. 1988; Coome et al. 1990; Caldiera and Basili 1991; Cheng 1993

Table 1 Summary of Reuse Issue Research

Reusability - Benefits and Challenges

A successful reuse program usually provides benefits in three areas:
• increased productivity and timeliness of the software development process;
• improved quality of the software development product; and
• increased overall effectiveness of the software development process (Caldiera & Basili 1991; Karlsson 1995; Yongbeom & Stohr 1998; Thorne 1999).

A number of success stories, detailing the benefits that organisations have received from implementing reuse programs, have been published. Examples include GTE Data Services saving $1.5 million on the Asset Management Program by applying systematic reuse (Prieto-Diaz 1993), and Raytheon's Missile Systems Division, Information Processing Systems Organisation, where a disciplined reuse program helped the organisation increase productivity by 60 per cent and also allowed the organisation to improve product quality and process flexibility (Haley 1996).

Even though there is little empirical evidence to support claims of productivity increases from reuse, there are reported results from many software reuse projects that show increases in productivity due to systematic application of reuse (Selby 1989; Coome et al. 1990; Frakes et al. 1991). Moreover, an increase in productivity in the order of 20 to 40 percent seems most common (Poulin et al. 1993). For the United States Department of Defense, Boehm (1999) estimates a 47 percent productivity boost could be gained by implementing a reuse strategy.

The increase in software quality with reuse comes about because reusable components tend to be thoroughly tested and are defect-free (Swanson et al. 1991; Lim 1994). Other advantages obtained from high quality software are adaptability, portability and understandability (Karlsson 1995; Yongbeom & Stohr 1998). Regarding the overall software development process, reuse provides flexibility, predictability and efficiency, allowing software firms to attack new markets though cost reduction and differentiation (Griss et al. 1994).

Even though reuse provides various benefits, there are two main challenges that are associated with large-scale reuse programs: cost; and management commitment (Apte et al. 1990; Yongbeom & Stohr 1998; Thorne 1999).

At the beginning of the reuse program itself, costs are involved to find, understand, adapt and integrate resources in the final product. Then, ongoing investment is required to maintain the reuse library and develop software resources for reuse. The other challenge with large-scale reuse projects is convincing management to commit and invest time and money into the reuse program. This managerial issue is a bigger challenge than the technical difficulties facing reusability and is critical in ensuring the success of a reuse program (Apte et al. 1990; Karlsson 1995; Boehm 1999).

SOFTWARE PROCESS PERFORMANCE DIMENSIONS

Software process performance is defined as the actual results achieved by following a software process (Paulk et al. 1995). Software process performance is usually measured using three dimensions: flexibility; productivity; and predictability (Cusumano 1991; Nidumolu & Knotts 1998).

Process Flexibility

The term *process flexibility* is often used in the manufacturing industry to express the ability to build different types of products in the same manufacturing plant or on the same production line at the same time (Browne et al. 1984; Jordan & Graves 1995). In the manufacturing industry, flexibility is important because firms need to respond quickly to changing market demands or risk losing market share due to fierce competition (Johnson 1990; Cusumano 1991; Parthasarthy & Sethi 1992; Jordan & Graves 1995).

In the software industry, the issue of flexibility is now becoming increasingly important considering the pressure on software firms to reduce time-to-market due to an increasing demand for software products. Consequently, and also partly due to an increased awareness of process-based approaches to software development, process flexibility has now become a point of focus in MIS research (e.g. Parthasarthy & Sethi 1992; Bantel 1993; Olsen 1995; Jordan & Graves 1995; Nidumolu & Knotts 1998).

Process flexibility is defined as the speed with which the organisation's software development approach can respond effectively to changes in the organisation's environment. It is an important strategy for dealing with the uncertainties in the current business environment. Researchers such as Olsen (1995) argue that software firms should invest in process flexibility by automating their software production lines and by using reusable components from previous projects in order to increase their market responsiveness and cost effectiveness.

Process Productivity

Productivity is defined in its simplest form as the ratio of outputs to inputs (Banker & Kauffman 1991; Lim 1994). In the manufacturing industry, techniques such as customisation, flexible automation, and reuse have been employed to increase productivity. In software development, productivity is critical given the increasing need to deliver high quality software on time and on budget. Productivity is often broken down into two dimensions: *efficiency* and *effectiveness* (Banker & Kauffman 1991; Stevenson 1995). Efficiency is concerned with the resources consumed in producing a given application in a timely manner, whereas effectiveness is concerned with the quality of the finished products and its appropriateness to the initial problem (Scudder & Kucic 1991).

Considering the current dynamic and uncertain business environment, it is important for software firms to respond quickly to the market with low-cost products. However, at the same time, software organisations should ensure that products are of high quality and hence meet customer requirements. Thus, in capturing the full concept of productivity, it is important to consider both efficiency and effectiveness.

Process Predictability

A stable and predictable manufacturing process has been critical to mass producers (Boynton et al. 1993). In software development, process predictability is becoming a necessity for the survival of software firms, given the need to deliver products within budget, and on time.

The maturity of an organisation's software process helps to predict a project's ability to meet its goals. According to Paulk et al. (1995), projects in Level 1 organisations (based on the CMM) experience wide variations in achieving cost, schedule, functionality, and quality targets. In order to advance to the next maturity level, these firms need to be able to meet their cost, time and performance targets. In this study, the definition of process predictability was adopted from Dowson (1993) who defines *process predictability* as the ability of the software firm to accurately estimate the needed resources, time, performance, quality and functionality of its software projects.

DEVELOPMENT AND APPLICATION OF THE RESEARCH MODEL

Recapitulating from the literature that has been discussed above, there is an increasing need to improve competitive performance of software firms. Based on the discussion, it was found that the most effective way to address the challenge of increasing competitive performance of software firms is through the key objective of process-based approaches to software development: reusability. Reusability helps software firms be more competitive by improving the firm's software process performance, i.e. flexibility, productivity and predictability. In Figure 1, the research model shows the relationship between the key variables investigated in this study.

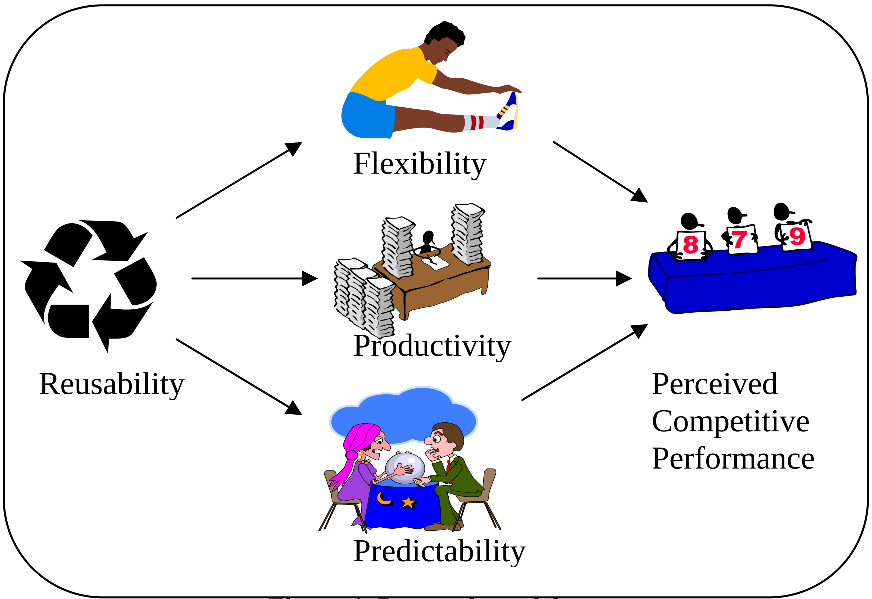

Figure 1. Research model

Survey

In the past, there has been a tendency by IS researchers to use a single item measure for the dependent variable, rendering the results purely speculative. To overcome this risk, DeLone and MacLean (1992) emphasise the importance of developing multi-dimensional measures. To adequately and accurately measure a complex variable such as perceived competitive performance, a multi-dimensional measure is needed (Nidumolu and Knotts 1998).

Based on the research model and previously validated research instruments, a questionnaire was developed. Multi-dimensional measures of perceived competitive performance were determined from the competitive performance literature. The multi-item measures of the independent variable (reusability) and the intermediate variables (process flexibility, process productivity, process predictability) were also developed from previously developed and validated IS research instruments as shown in table 2. The items for perceived competitive performance, reusability, process flexibility and process predictability were borrowed from the study of Nidumolu and Knotts (1998). These items were compiled from previous research done by researchers in both the manufacturing industry and the software industry. The items for process productivity on the other hand were mainly modelled on the study by Lane (1998).

Construct	Meaning in Previous Studies	Authors
Perceived Competitive Performance	Market Responsiveness Product Cost Efficiency	Nemetz & Fry 1988; Boynton et al. 1993; Pine 1993; Nidumolu & Knotts 1998* Nemetz & Fry 1988; Johnson 1990; Boynton et al. 1993; Nidumolu & Knotts 1998*
Process Flexibility	Speed of response to the business environment	Parthasarthy & Sethi 1992; Bantel 1993; Nidumolu & Knotts 1998*
Process Productivity	Efficiency Effectiveness	Banker et al.1991; Henderson & Lee 1992; Deephouse et al. 1995; Lane 1998* Krieg & Gosler 1989; Deephouse et al. 1995; Lane 1998*
Process Predictability	Ability to accurately predict time and resources before a project start	Boynton et al. 1993; Henderson & Lee 1992; Keller 1994; Nidumolu & Knotts 1998*
Reusability	Reuse of 'anything' throughout the system development	Basili 1988; Apte et al. 1990; Nidumolu & Knotts 1998*

* Study where measuring items were borrowed

Table 2 Sources of Measurement Instruments

For the survey, the target population was identified as all commercial organisations in Queensland and New South Wales which develop package or custom software. The Yellow Pages Telephone Directory was used to extract a list of 728 organisations operating in Queensland or New South Wales, and belonging to the category: 'develop or sell computer software or hardware, or provide computer consulting'. This target sample was reduced to 375 by eliminating any organisation dealing solely in computer hardware or consultancy. After pre- and pilot testing to maximise reliability and validity of the measuring instruments, the questionnaire was mailed to 320 organisations randomly drawn from the 375 in the target sample. Although a total of 118 questionnaires were returned, only 46 of these were valid and usable. Temporal and financial constraints limited further follow up.

Findings

The vast majority (93%) of respondents represented small organisations with less than 10 full-time software developers and a budget/turnover of less than $5 million. Most (78%) have been in operation for at least five years and represented mainly retail (22%) education (17%) and finance and insurance (17%) sectors of the economy. The findings confirm that reuse has been around in Australia for at least 15 years, with a high proportion (67%) of respondents reporting at least 5 years reuse experience. The popularity of the five mechanisms suggested to promote reuse were varied. As can be seen in Table 3, the most accepted mechanisms are establishing a library of reusable resources and the inclusion of reusability as a formal part of the software development approach. In considering software productivity measures, structured design (54%) and time and cost (67%) were used more than other measures.

Mechanism to promote reuse	Used in at least some projects
Library of reuse resources	89%
Reusability as formal part of software development approach	67%
Reusability as formal part of employee appraisal	41%
Reward system for reusing existing resources	26%
Reward system for creating reusable resources	24%

Table 3. Mechanisms adopted to promote reuse

Factor analysis was performed on the 22 items representing the three process performance dimensions (process flexibility, process productivity and process predictability). In the data reduction stage, some of the items were removed and some loaded under different variables. The new process performance variables were determined:

- *techno-regulatory flexibility* - the ability to respond quickly when new software or hardware becomes available, and when laws regulating business are enacted or repealed;
- *process efficiency* - the ability to predict required resources and use them efficiently;
- *process effectiveness* - predicted and actual quality of software product; and
- *labour flexibility* - the ability to respond to changes in quality or quantity of software developers.

The revised relationships were then analysed using simple and multiple regression analysis to determine the effect of reusability on perceived competitive performance through its effect on the four new software performance dimensions (results provided in Table 4). Due to the low response rate, the statistical power was considered to be poor.

Software Process Performance Variables	Reusability		Market Responsiveness		Product Cost Efficiency
	F-value	Adjusted R^2	Beta value	t-value	t-value
Techno-Regulatory Flexibility	0.043*	0.071	0.484	0.001*	0.880 (ns)
Process Efficiency	0.050*	0.065	0.463	0.162 (ns)	0.272 (ns)
Process Effectiveness	0.837 (ns)	0.001	0.093	0.490 (ns)	0.549 (ns)
Labor Flexibility	0.244 (ns)	0.009	0.116	0.389 (ns)	0.856 (ns)
			Adjusted R^2	0.222	-0.056

* $p < 0.05$ ns = not significant

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

Table 4 - Results of simple and multiple regression analysis

Reusability and the Software Process Performance Dimensions

With rapid changes in laws (e.g. GST) and technologies impacting the software industry, firms need to be more vigilant and 'techno-regulatory' flexible to remain competitive. A weak relationship was confirmed between reusability and techno-regulatory flexibility, supporting prior research (by Cusumano 1991; Swanson et al. 1991; Reifer 1992) that reusability helps improve process flexibility by reducing development time and facilitating software modifications.

A weak relationship was also confirmed between reusability and process effectiveness, but possibly due to the small size of the respondent firms, no significant relationship was found between reusability and process efficiency or labour flexibility. In very small firms, the labour turnover is usually low and stable so that changes in quantity and quality of software developers (labour flexibility) would not have a major effect.

Software Process Performance Dimensions and Perceived Competitive Performance

The study identified two dimensions of perceived competitive performance: market responsiveness and product cost efficiency. The relationship between techno-regulatory flexibility and market responsiveness was confirmed, consistent with previous research (e.g. Bantel 1993; Boynton et al. 1993; Nidumolu & Knotts 1998). Organisations with flexibility in response to changing regulations and technology are able to introduce new products ahead of their competitors.

However, for the three other process performance dimensions (efficiency, effectiveness and labour flexibility), there was no support for their relationship with market responsiveness. Perhaps other factors outside the scope of this research, such as project size, creativity, planning formality and environmental complexity had a confounding impact on the model.

As mentioned previously, most of the respondents represented very small firms to whom labour flexibility may not be important.

For the other dimension of perceived competitive performance, product cost efficiency, no support was found for the relationship between it and any of the process performance dimensions. Once again, this could be related to the high proportion of small respondent firms. Many small firms find it difficult to quantify the benefits received from reusability (Broadman & Johnson 1997).

CONCLUSION

This paper has presented a research framework to investigate the relationship between reusability, software process performance (flexibility, productivity and predictability) and perceived competitive performance. 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. However, caution needs to be exercised in relation to the findings as the low response rate imposes limitations on the statistical validity and generalisability of results.

Since this study involves a new area of research and the measurement instruments are relatively new, further research could refine the measurement instruments by conducting a number of case studies, thereby increasing the validity and reliability of the research variables. A survey across all Australian states would then provide the opportunity to generalise results and provide a better insight into the Australian software industry's adoption of reuse strategies. By providing practitioners with accurate information about the benefits and risks of reuse and its potential to improve software process performance, management may be convinced to provide the necessary commitment to adopt reuse strategies. In the long term, this research may help Australian software firms to achieve competitive performance and compete successfully in the global market.

REFERENCES

ABS (1997) *Computing Services Industry, Australia 1995-96,* Australian Bureau of Statistics, Canberra.

Apte, U. Sankar, C.S. Thakur, M. and Turner J.E. (1990) Reusability-Based Strategy for Development of Information Systems, *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.

Basili, V.R. (1988) "Towards a Comprehensive Framework for Reuse", *Tech. Report CS-TR-2158,* C.S. Dept., Univ. of Maryland, College Park, Md.

Boehm, B. W. (1987) *Improving Software Productivity,* IEEE Computer Society Press.

Boehm, B.W. (1999) Managing Software Productivity and Reuse, *Computer*, September.

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 CMM[SM] for Small Businesses, Small Organisations, and Small Projects*, LOGOS International Inc.

Browne, J., Dubois, D., Rathmill, K., Sethi, S.P., and Stecke, K.E. (1984) Classification of Flexible Manufacturing Systems, *The FMS Magazine,* April, 114-117.

BSA (Business Software Alliance) (1998) *The Contribution of Packaged Software Industry to the Australian Economy*, December.

Caldiera, G. and Basili, V.R. (1991) Identifying and Qualifying Reusable Software Components, *IEEE Computer*, 24(2), 61-70.

Cheng, J. (1993) Improving Software Reusability in Object-Orientated Programming, *ACM SIGSOFT Software Engineering Notes,* 18(4), 70-74.

Coome, T.N., Comer, J.R. and Rodjak, D.J. (1990) Developing Reusable Software for the Military Systems, *ACM SIGSOFT Software Engineering Notes*, vol.15, no.3, 33-38.

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).

DeLone, W.H. and MacLean, E.R. (1992) Information Systems Success: The Quest for the Dependent Variable, *Information Systems Research*, 3(4), 60-65.

Dowson, M. (1993) Software Process Themes and Issues, *Second International Conference on the Software Process*, Berlin, 54-62.

Due, R.T. (1995) The Economics of Reuse, *Information Systems Management*, 12(1).

El Emam, K. Drouin, J. and Walecio M (1998) SPICE: *The Theory and Practice of Software Process Improvement and Capability Determination*, IEEE Computer Society Press, Los Alamotos, California.

Frakes, W.B., Biggerstaff, T.J., Prieto-Diaz, R., Matsumura, K. and Schaefer, W. (1991) Software Reuse: Is it Delivering?, *Proc. 13th Int Conf Software Engineering*, May.

Golder, P.N. and Tellis, G.J.(1993) PioneerAdvantage: Marketing Logic or Marketing Legend?, *Journal of Marketing Research*, 30 158-170.

Griss, M.L., Favaro, J. and Walton, P. "Managerial and Organisational Issues – Starting and Running a Software Reuse Program" in Schaefer, W., Prieto-Diaz, R. and Matsumoto, M. (eds), 1994, *Software Reusability,* New York: Ellis Horwood, 51-78.

Haley, T.J. (1996) Software Process Improvement at Raytheon, *IEEE Software*, 13(6).

Henderson, J. C. and Lee, S. (1992) Managing I/S Design Teams: A Control Theory Perspective, *Management Science*, 38(6), 757-777.

Hooper, J.W. and Chester, R.O. (1991) *Software Reuse,* Plenum, New York.

Humphrey, W.S. (1989) *Managing the Software Process*, Addison-Wesley Publishing Company, New York.

Humphrey, W.S. Snyder, T. and Willis, R. (1991) Software Process Improvement at Hughes Aircraft, *IEEE Software*, July, 11-23.

Johnson, H.J. (1990) Performance Measures for Competitive Excellence, in *Measures for Excellence,* eds Kaplan, Harvard Business School Press, Boston.

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.

Keller, R.T. (1994) Technology-Information Process Fit and Performance of R&D Project Groups: A Test of Contingency Theory, *Academy of Management Journal*, 37(1).

Krieg, R.E. and Gosler, M.D. (1989) Integrating Technology to Increase Application Development Productivity, *Journal of Systems Management*, 40(8), 6-14.

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.

Love, T. (1988) The Economics of Reuse, *Proceedings of IEEE COMPCON*, 238-241.

Mayer, J.H. (1997) Swedish Reap Paybacks in Software Reuse Projects, *Military and Aerospace Electronics*, 8(4), 9-10.

McIlroy, M.D. (1968) Mass-Produced Software Components, *Software Engineering Concepts and Techniques:* 1968 NATO Conference Software Engineering, Germany, 88-98.

Mili, H., Mili, F. and Mili, A. (1995) Reusing Software: Issues and Research Directions, *IEEE Transactions on Software Engineering*, 26(6), 528-561.

Nemetz, P.L and Fry, L.W. (1988) Flexible Manufacturing Organisations: Implications for Strategy Formulation and Organisational Design, *Academy of Management Review*, 13(4), 627-638.

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.

Olsen, N.C. (1995) Survival of the Fastest: Improving Service Velocity, *IEEE Software*, Sept.

Parthasarthy, R. and Sethi, S.P. (1992) The Impact of Flexible Automation on Business Strategy and Organisational Structure, *Academy of Management Review*, vol.17.

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.

Pine, B. (1993) *Mass Customisation: The New Frontier in Business Competition*, Harvard Business School Press, London.

Porter, M. E. (1985) *Competitive Advantage – Creating and Sustaining Superior Performance*, Collier Macmillan Publishers, London.

Poulin, J.S., Caruso, J.M. and Hancock, D.R. (1993) The Business Case of Software Reuse, *IBM System Journal*, 34(4), 567-594.

Prieto-Diaz, R. (1993) Status Report: Software Reusability, *IEEE Software*, May, p.52.

Ramamoothy, C.V., Garg, V. and Parkash, A. (1988) Support for Reusability in Genesis, *IEEE Transactions on Software Engineering*, 14(8), 1145-1154.

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.

Rubin, K. (1990) Reuse in Software Engineering: An Object-Orientated Perspective, *Proceedings of IEEE COMPCON*, 340-346.

Scudder, R.A. and Kucic, A.R. (1991) Productivity Measures for Information Systems, *Information & Management*, 20(5), 343-354.

Selby, R.W. (1989) "Quantitative Studies in Software Reuse", in Biggerstaff, T. & Perlis, A.J. 1989, *Software Reusability*, Addison-Wesley, New York.

Stevenson, C. (1995) *Software Engineering Productivity*, Chapman & Hall, Melbourne.

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), Feb.

Vendelo, M.T. (1998) Recycling Software – On the Road to High Performance in Software Companies, *International Journal Technology Management*, 16(1/2/3).

Ward, J. and Griffiths, P. (1998) *Strategic Planning for Information*, John Wiley, Chichester.

Wohlwend, H. and Rosenbaum, S. (1993) Software Improvements in an International Company, *Proc. 15th ICSE*, IEEE Computer Society Press, Los Alamitos, California.

Woods, W.A. (1999) Still No Silver Bullet: Managing and Improving the Software Development Process, *Armed Forces Comptroller*, 44(1), 33-37.

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 & Fall of the American Programmer*, Yourdon Press Englewood Cliffs, New Jersey.

COPYRIGHT