ICIS 2001 Proceedings

International Conference on Information Systems (ICIS)

December 2001

The Relationship of Software System Flexibility to Software System and Team Performance

Kay Nelson
Ohio Sate University

Jay Cooprider
Bentley College

THE RELATIONSHIP OF SOFTWARE SYSTEM FLEXIBILITY TO SOFTWARE SYSTEM AND TEAM PERFORMANCE

Kay M. Nelson
Fisher College of Business
The Ohio State University
Nelson_k@cob.osu.edu

Jay G. Cooprider
Trustee Professor
Computer Information Systems
Bentley College
JCooprider@bentley.edu

Abstract

Organizations are evolving at an ever-increasing rate due to economic changes, globalization, and technology enabled changes such as e-business. These changes require existing and new software systems to be flexible enough to support rapid change while still being reliable and cost effective. Software flexibility has previously been studied only from a structural perspective with little attention to the people and processes that support the software, especially in the maintenance environment where most changes occur. This research builds on a previously validated technology flexibility model with two dimensions: structural and process flexibility (Nelson and Ghods 1998). We empirically test the relationship of flexibility to the performance of the software system and the IS support team. The relationships between flexibility and performance are analyzed using a "fit" analysis to fully understand the complexity of the relationships. Our findings indicate that software flexibility is related to both system and team performance, and is characterized by the amount of interaction, or "fit," between structural and process flexibility.

Keywords: Flexibility, maintenance, team performance, system performance.

INTRODUCTION

Flexibility has become a key characteristic desired in both software systems and business processes (Davenport 1992; Harrington 1991). The quest for flexibility is driven by shorter product cycle times, global competition, constant demands to reduce and control costs, the need to answer to shareholder demands, and moves to connect organizations through e-business initiatives (Markus 2000). In an attempt to respond to the need for flexibility, organizations typically deploy new "flexible" software systems such as enterprise resource planning systems (ERP) or attempt to change existing systems (Allen and Boyton 1991; Byrd and Turner 2000). Whichever option is chosen, these applications are required to continually adapt during their operating life. This study examines software flexibility during the maintenance phase of the lifecycle, where most changes are made to a system.

FLEXIBILITY

Not all software needs to be flexible. Structural and organizational flexibility are investments that should be cost justified by the business strategy. Previous research has shown flexibility to have "little penalty in time, effort, cost, or performance" (Upton 1995, pg. 73). Organizations must choose to make investments in software system flexibility, just as they choose to invest in speed or reliability. Technology is often used to gain manufacturing flexibility (Gerwin 1993; Upton 1994). In a study of 61 North American paper mills, managers rated 40% of their flexibility improvement efforts to be unsuccessful (Upton 1995). The primary cause of these failures was the reliance on technology alone to provide flexibility. It was the interaction of technology and people that produced flexibility in successful projects, suggesting that the same may be true for software system flexibility.

We use definitions of software system flexibility derived from Huber and McDaniel's (1986) definition of organizational flexibility. Combining their definitions with the findings of Upton (1995), **software system flexibility** is defined as:

> The ease of changing a software system's structure and processes with little penalty in time, effort, cost, or performance.

Software system flexibility is a two-dimensional construct composed of structural and process flexibility.

> **Structural flexibility** is the capability of the design and organization of a software application to be successfully adapted to business changes.
>
> **Process flexibility** is the ability of people to make changes to the technology using management processes that support business changes.

The determinants of structural and process flexibility are based on measures of flexibility in the behavioral psychology and software engineering literature (Table 1) (Nelson and Ghods 1998).

Table 1. Determinants of Structural and Process Flexibility

Determinant	Definition
Structural Flexibility	
Modularity	The degree of formal design separation within a technology
Change Acceptance	The degree to which a technology contains built-in capacity for change
Consistency	The degree to which data and components are integrated consistently across a technology
Process Flexibility	
Rate of Response	The degree to which changes can be made to a technology in a timely manner
Expertise	The degree to which up-to-date knowledge about the operation and maintenance of a technology exists and is communicated
Coordination of Action	The degree to which technology maintenance and user organizations operate according to the requirements of each other and the total organization

IS PERFORMANCE

Previous researchers have used measures of both product and process in the measurement of IS performance (Cooprider and Henderson 1989). This research uses system performance as a measure of product performance (Gibson and Senn 1989) and team performance as a measure of the process of maintaining the software system (Hyman 1993). The measures used were reported by IS and user managers who are stakeholders of both the software application and the software maintenance team.

We conceptualize performance as having two dimensions: **system performance** and IS maintenance **team performance**. The system performance dimension captures the technical, production aspect of performance. The team performance dimension uses an organizational approach to capture the process aspect of IS performance (Hyman 1993). These dimensions capture the performance of both the software application and the people who maintain the application, consistent with our prior definition of a software system.

The indicators of system performance and team performance were drawn from previously validated measures used in multiple studies, and based on performance characteristics deemed important by stakeholder respondents during the pilot test phase of this study. System performance is conceptualized with three indicators: overall quality, ease of use, and timeliness of data delivery. The quality of a software application has been found to be a reliable indicator of future system performance in many types of systems (Basili and Briand 1996; DeLone and McLean 1992;King and Epstein 1983). Ease of use has been used as an indicator of both user satisfaction and the overall impact of the system on the organization (DeLone and McLean 1992). We have chosen ease of use as an indicator of system performance since it has been repeatedly shown to lead to usage of the system, which will impact system performance (Hamilton and Chervany 1981). The timeliness of data delivery indicator captures the information quality aspect of system performance (DeLone and McLean 1992). While many other indicators of data quality could have been chosen, we believe that in an investigation of flexibility, the ability of data delivery to keep up with the business is a critical indicator of system performance (Allen and Boynton 1991; King and Epstein 1983). The above determinants of system performance are derived from the technical performance of the system. Therefore, we hypothesize that:

H1: Structural flexibility is directly related to system performance.

The indicators of IS team performance were drawn from both discovery interviews and the IS literature. The majority of managers interviewed, both IS and business, cited ability to adhere to budget and schedule as primary measures of IS team performance. These indicators have also been widely used in the IS literature as dependent measures of IS team performance (DeLone and McLean 1992; Guinan et al. 1998; Miller and Doyle 1987). While reputation has not been as widely used in the IS literature as a dependent measure of team performance, the stakeholders in this study were cited as an important indicator of IS team performance. We believe that reputation is an especially appropriate indicator when subjectively measuring team performance from the stakeholder perspective (Guinan et al. 1998).

While some systems are designed for ease of maintainability, the software engineering literature has widely shown that maintenance team performance is a result of the process discipline of that team (Humphrey 1995; Paulk et al. 1993). Therefore, we do not expect structural flexibility to be significantly related to team performance.

H2: Structural flexibility is not directly related to IS maintenance team performance.

Many IS organizations are currently using process improvement techniques and certification to improve performance (Humphrey 1995; Paulk et al. 1993; Sommerville 1989; Yourdon 1986). These techniques are designed to improve both product and process performance. Therefore, we hypothesize that the process flexibility dimension of software system flexibility will be related to both system and team performance.

H3: Process flexibility is directly related to system performance.

H4: Process flexibility is directly related to team performance.

Based on Huber and McDaniel's (1986) work on organizational flexibility, we expect that there will be some interaction between structural and process flexibility that will impact performance. Systems theory (Thompson 1967) suggests that this interaction should have a favorable impact on performance, so we hypothesize that interaction between the dimensions will be more strongly related to performance.

H5: Software systems that have a higher degree of interaction between **structural flexibility** and **process flexibility** have a stronger relationship between **flexibility** and **performance** than systems without a high degree of interaction.

These interaction effects will be assessed using tests for "fit" as covariance, "fit" as moderation, and "fit" as matching as recommended by Venkatraman (1989).

RESEARCH DESIGN AND METHODS

Since this research is both theory building and theory testing, validity is a primary consideration in the analysis process. This study uses Bagozzi's (1980) criteria for testing construct validity. Bagozzi's tests of validity are theoretical and observational meaningfulness of concepts, internal consistency of operationalizations, and convergent, discriminate, and nomological validity.

This paper uses the previously validated technology flexibility model (Nelson and Ghods 1998) (Table 1) to test whether the software flexibility determinants are related to IS performance. A combination of qualitative and quantitative techniques was used to derive and test the models used in this study. Candidate questions for both the flexibility and performance models were drawn from extensive discovery interviews (Rossi et al. 1983).

A pretest sort of these indicators was then conducted to further strengthen observational meaningfulness. A total of 23 pilot questionnaires were collected from respondents in four organizations. Each pilot question was reviewed for content, clarity, and meaning. Through this process, candidate questions were further refined and selected for inclusion in the final research instruments. Based on the results of the pilot test, at least four questions for each determinate of flexibility, and three for each performance construct, were retained for the survey phase of the study. The criteria used for keeping questions were clarity, meaningfulness, ability to measure the construct, and understandability. This technique was employed to test observational and theoretical meaningfulness of constructs.

This study is a cross sectional field study. The sample was a heterogeneous set of representative organizations and software systems. This sample was drawn from 12 organizations chosen for industry diversity, ease of data collection, and availability of metrics information, making this a convenience sample. All determinants were measured on a seven-point Likert scale survey instrument. Respondent groups represented 116 software systems across the 12 organizations, each built and maintained by distinct teams. For each represented software system, flexibility data was collected from expert respondents: one or two maintainers and one or two users of the system. Performance data was collected from stakeholders of these systems to avoid self-report bias on performance indicators. The data was analyzed at the group level of analysis, aggregating responses that represented each software system. The respondents for the study averaged 41.3 years in age, had an average of 15.3 years IS or IS related (in the case of the users) experience, an average 4.7 years experience with the application on which they were reporting, and were 31.9% female.

ANALYSIS

Reliability

The first step in testing whether software system flexibility is related to software system and support team performance is to analyze the reliability of the determinants proposed in Table 1. This research utilizes Joreskog's (1974) analysis of covariance structures, which tests both unidimensionality and reliability simultaneously. This reliability assessment is recommended by Bagozzi (1980) over the traditional Cronbach alpha reliability test because it does not assume that all indicators are equally important. The results show the constructs to be reliable. Both construct reliabilities (system and team) are above 0.70, so they easily fall into the acceptable range ($\rho > 0.50$) specified by Bagozzi. [Complete statistical results are available on request from the authors.]

Validation of Performance Model

The next step tests whether software system flexibility is related to the performance model. This validation was performed using structural equation modeling (with LISREL) and comparing alternate models (Anderson and Gerbing 1988). The results of this analysis are shown in Table 2.

Table 2. Summary of Results
Performance Measurement Models

Model	χ^2	df	*p*-level	BBI	RMSEA	GFI	AGFI
P^*_n	212.89	15	< .001	n/a	.35	.52	.33
P_1	41.90	9	< .001	.80	.19	.87	.69
P_2	9.17	9	.33	.96	.04	.97	.93
P_C	31.09	9	< .001	.92	.15	.92	.82

Model P_n is the null model, and posits that each of the six indicators measures a separate independent factor. Model P_1 assumes that all six indicators share a single underlying factor. Model P_2 is the theorized performance measurement model, proposing two underlying factors of performance. Model P_C is a constrained model where the correlation between the two constructs of performance is set to one. This model is used to establish discriminant validity by comparing it to the measurement model of performance (Model P_2). Using LISREL CFA, a high *p* level indicates a lower likelihood that a better fitting model exists. For each of the models, the table shows the χ^2 goodness of fit statistic, the degrees of freedom for the statistic, the *p* value, the Bentler and Bonnet normed index (BBI), the root mean square error of approximation (RMSEA), the goodness of fit (GFI), and the adjusted goodness of fit (AGFI) (Bentler and Bonnett 1980; Burme 1989). Using LISREL CFA, a high *p* level indicates a lower likelihood that a better fitting model exists. The BBI is an indicator of the practical significance of the model in explaining the data. The rule of thumb is that BBI should be greater than 0.90 (Bentler and Bonnett, 1980). The GFI and AGFI provided by the LISREL program measure how much better a model fits the data than no model at all. AGFI should be .90 or better.

The analysis shows that P_1 explains about 80% of the variation in the data while P_2 explains 96% of the variation. P_2 provides an excellent fit with the data ($p > .1$), shows excellent RMSEA, GFI and AGFI results, and is significantly better than P_1. We therefore conclude that the theorized two-dimensional model, P_2, best explains performance. This analysis also shows that the χ^2 value for the unconstrained model (P_2) is $9.17_{(df:9)}$, while the χ^2 for the more constrained model (P_C) is $31.09_{(df:9)}$. The difference $\chi^2_{(df:0)}$ is 22.73, which is highly significant ($p < 0.001$), providing strong evidence for discriminant validity. In addition, the factor loadings on the constructs in model P_2 were all large and significant. This is evidence that each determinant is strongly related to its underlying theoretical construct, and demonstrates convergent validity.

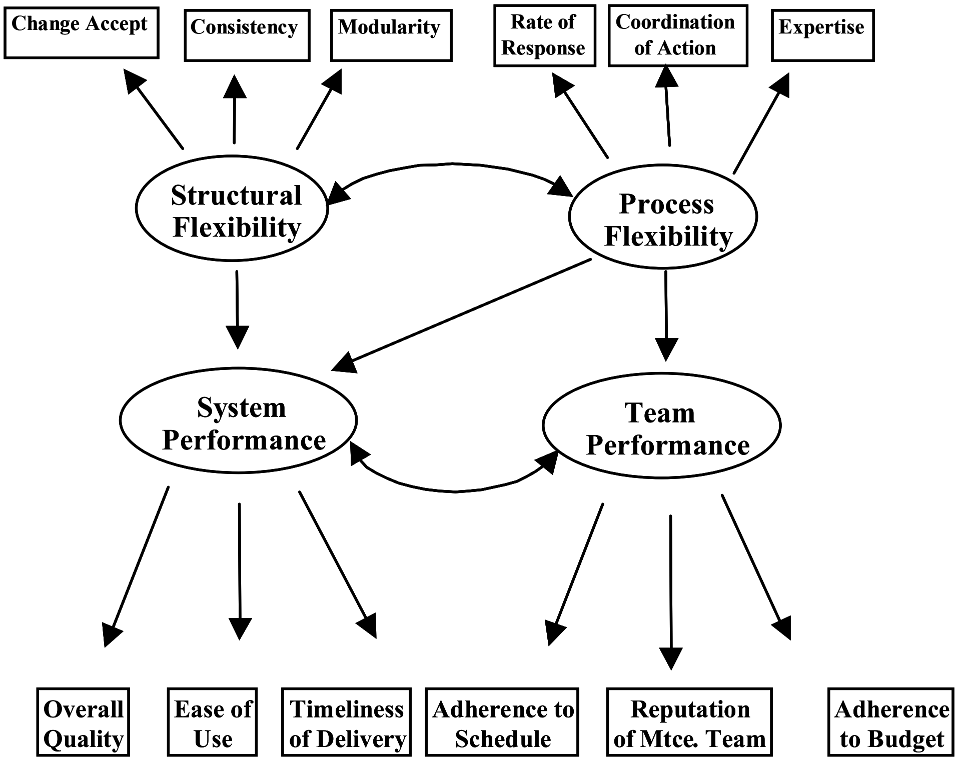

Figure 1. Structural Model of Software System Flexibility and Performance Model S_u

Predictive Validity of Performance Model

The comparison of alternate nested models approach (Anderson and Gerbing 1988) was also used to test the nomological or predictive validity of the model. Table 3 illustrates that model S_u, which allows all of the relationships between software system flexibility and performance to be estimated, is the model that provides the best explanation of the software system flexibility and performance data.

Model S_u shows that process flexibility has a significant (t = .05) relationship to system performance and team performance, but that structural flexibility only relates to system performance. This finding supports Hypotheses 1, 2, 3, and 4.

Table 3. Summary of Results
Predictive Validity Test

Model	χ^2	df	*p*-level	BBI	RMSEA	GFI	AGFI
S^*_n	680.70	66	< .001	n/a	.32	.50	.30
S_u	64.46	48	.06	.91	.06	.92	.86
S_c	70.07	50	.03	.90	.03	.91	.86

Fit and the Relationship Between Software System Flexibility and Performance

Hypothesis 5 states that a higher degree of interaction between the flexibility constructs will have a stronger relationship to performance than when no interaction is present. The lower goodness of fit of the constrained model S_c (Table 3) compared to model S_u indicates that the issue is not simply one of fit among all the first-order factors. We, therefore, propose fit as second-order factor reflecting the co-alignment of structural and process flexibility, or fit as covariance (Venkatraman 1989). This model (F_1) is tested using confirmatory factor analysis (CFA). In model F_1, the covariances between factors is set to zero and are then compared to a model where the factors are allowed to covary (F_2). In addition, the variances of the software system flexibility dimensions in F_1 are set equal to each other and the variances of the performance dimensions are set equal. This is done to provide identified variances to the model. The constraint of equality is based on the fact that each of these pairs of first order latent variables is derived from indicators measured with the same scale on the same instrument (Lord and Novick 1968).

Table 4. Summary of Results
Fit as Covariance

Model	χ^2	df	*p*-level	BBI	RMSEA	GFI	AGFI
F_1	78.39	52	< .01	.88	.07	.90	.85
F_2	64.46	48	.06	.91	.06	.92	.86

Table 4 shows that F_2 is a better model than F_1. Looking more closely at Model F_1, we find that the covariances between the software system flexibility and performance variables are negative and non-significant with the exception of the covariance between process flexibility and system performance, which is significant. This also provides evidence that an interaction effect exists among the software system flexibility and performance variables, but that this relationship is more than just fit as covariance.

Fit as moderation and fit as matching were tested using moderated regression analysis (MRA) (Venkatraman 1989). The factor loadings obtained in the models are used to calculate four values: (1) the level of structural flexibility, (2) the level of process flexibility, (3) the level of system performance, and (4) the level of team performance. A software system flexibility "interaction score" is obtained by multiplying the calculated values of structural and process flexibility ("Structure * Process") to test "fit as moderation," and a "deviation score" is obtained by taking the absolute value of the difference between the calculated values (|Structure - Process|) to test "fit as matching."

We test for both the *form* and the *strength* of the relationship between structural and process flexibility (Arnold 1982). The *form* of the relationship is tested through the use of MRA. Testing the form of the relationship proposes that the performance outcome is jointly determined by the interaction of structural and process flexibility (Arnold 1982; Venkatraman 1989). Examining the differences in the correlation coefficients between the constructs across subgroups tests the strength of the relationship. If the value of the correlation coefficient (r_{xy}) differs significantly across the subgroups, it supports that fit affects the *strength* of the relationship (Venkatraman 1989). The significance of differences in correlation coefficients is tested using the Fisher *z* transformation (Arnold 1982; Glass and Hopkins 1984). All significant relationships in these analyses are at the $p < .05$ level.

Table 5 summarizes the statistically significant findings of this study. [Statistics are available from the authors.]

Table 5. Summary of Research Findings

FINDINGS	ANALYSIS
Structural Flexibility is related to System Performance	Performance Model Su
Process Flexibility is related to both System and Team Performance	Performance Model Su
Process Flexibility is significantly related to System Performance in systems with high covariance levels of structure and process	Fit as Covariation
Structural Flexibility is related to System and Team Performance in systems with low levels of multiplicative interaction	Fit as Moderation
Structural and Process Flexibility are related to System and Team Performance in systems with high multiplicative levels of interaction	Fit as Moderation
Structural and Process Flexibility are more significantly related to System Performance in systems with high levels of interaction multiplicative than in those with low levels	Fit as Moderation
A close match of Structural and Process Flexibility is related to Team performance	Fit as Matching
Systems with the lowest match of the dimensions have Structural and Process Flexibility related to only System Performance	Fit as Matching
Systems with the highest match of the dimensions have Structural and Process Flexibility related to System and Team Performance	Fit as Matching
Process Flexibility is more significantly related to System Performance in systems with high levels of matching than in those with low levels	Fit as Matching

DISCUSSION AND SUMMARY

The results of our analyses indicate that the performance model of software system flexibility meets Bagozzi's (1980) criteria for validity. Model S_u supports Hypotheses 1 through 4. The results of the confirmatory factor analysis performed demonstrate that structural flexibility is directly related to system performance and that process flexibility is directly related to both system and team performance. Teams perform the way they do regardless of the structural flexibility of a software system, but the process flexibility of the system is related to their performance. The performance of the software application itself, the overall quality of the application, the ease of use of the application and the timeliness of data delivery by the application, are related to both dimensions of flexibility. For example, if the structural flexibility of a software system is viewed as high, this indicates that the system performance measures are more likely to be rated as high—but it provides no information concerning the team performance measures. However, if process flexibility is viewed as high, this is an indication that both system performance and team performance are more likely to be high. This result indicates that the process flexibility dimension of software system flexibility, the ability of people using management processes to make changes in the software application in support of business process changes, has more direct relationship to software system flexibility performance than the way a software system is structured.

The analysis of construct interaction fit reveals more insight into how structural and process flexibility relate to system and team performance, and confirms Hypothesis 5. As structural and process flexibility co-vary more closely, the only significant (.05) relationship is that of process flexibility to system performance. This means that as the amount of resources put into structural and process flexibility becomes more consistent, process flexibility will be related to system performance (Venkatraman 1989). This is a very interesting finding in light of Upton's (1995) findings that overall flexibility was not only a function of the technology, but of the people using the technology as well.

The analysis of fit as moderation assumed the flexibility constructs to have a multiplicative interaction, which did not have a relationship to performance when tested across all software systems. However, software systems with low levels of multiplicative interaction between structural and process flexibility have significant relationships between structural flexibility and both system and team performance. These systems show no relationship between process flexibility and either performance construct. It appears that when there is low interaction between structural and process flexibility, structural flexibility becomes the more important dimension in terms of performance. In the systems with high levels of structural and process flexibility interaction, structural and process flexibility are correlated to both performance constructs. A high level of multiplicative interaction between the constructs seems to lead to process flexibility being more closely related to performance.

The results of the fit as matching analysis were similar to those of the fit as moderation analysis. The only significant (.05) difference in correlations between the low match and high match systems was in the relationship of process flexibility to system performance. This is different from the analysis of fit as moderation where the significant difference was in the relationship of process flexibility to both system and team performance. The idea of match between constructs has some interesting implications. Systems with high levels of both structural and process flexibility have relationships to system and team performance, as do teams with low levels of both dimensions. It is the fact that the levels of structural and process flexibility are *in agreement,* rather than either high or low,that is significant. This finding can be interpreted in several ways. It is possible that system expectations are related to the level of flexibility required of a particular business process. If the business process supported by a software system is not required by the environment to be flexible, the software system that supports it may also not be required to be flexible. All software systems need not be equally flexible. What appears to be important to both system and team performance is that the levels of structural and process flexibility match. These results also reveal that systems with a mismatch between structural and process flexibility do not have relationships between the software system flexibility dimensions and team performance. One way of viewing this is that these systems are at the mercy of the technology. If the technology has high structural flexibility, the software maintenance team and the system perform well. If not, process flexibility does not give them an advantage.

It seems clear that in all cases except that of fit as covariation, a relationship exists between structural flexibility and system performance. The relationship of process flexibility to performance is not as clear. A significant difference exists between the systems with low and high levels of both moderation and matching in the relationship of process flexibility and system performance. With all of the analyses showing this relationship except that of the low interaction and matching systems, it seems likely that process flexibility must interact with and support structural flexibility to be related to system and team performance, with the relationship to system performance being the more significant of the two relationships. This attention to process flexibility has the potential to enhance structural flexibility that is built into software applications.

When we examine software applications with high levels of multiplicative interaction (fit as moderation) and the lowest amount of deviation between the flexibility constructs (fit as matching), we find that Hypothesis 4 is supported in that the relationships between the flexibility and both performance constructs is stronger. This indicates that attention to the people and management processes that support a software system may contribute to the overall flexibility and performance of the software system.

These findings suggest that management that desires performance through flexibility, but only invests in structural flexibility, may experience disappointment. Organizations seeking flexibility through software technology need to consider both the structural design of the software application and the people and processes used to support the application. By using this approach, organizations can explore several alternatives. Existing software can be examined to see if it can be modified for structural flexibility or if personnel can be given increased training to gain process flexibility. Software engineering techniques such as structured methodologies and certifications can be employed. It may be less expensive to gain flexibility by focusing on the process than by purchasing or developing new software. When a decision has been made to replace an existing system, the costs of gaining process flexibility to support structural flexibility should be considered in the overall project costs. It is possible that many of the traditional performance disappointments with software may result from a failure to consider process flexibility. The findings of this study indicate that a holistic view of flexibility contributes to both IS system and team performance.

References

Allen, B., and Boynton, A. “Information Architecture: In Search of Efficient Flexibility,” *MIS Quarterly* (15:4), 1991, pp. 435-446.

Anderson, J., and Gerbing, D. “Structural Equation Modeling in Practice: A Review and Recommended Two-Step Approach,” *Psychological Bulletin* (103:3), 1988, pp. 411-423.

Arnold, H. “Moderator Variables: A Clarification on Conceptual, analytic and Psychometric Issues,” *Organizational Behavior and Human Performance* (29), 1982, pp. 143-174.

Bagozzi, R. *Causal Modeling in Marketing*, Wiley & Sons, New York, 1980.

Basili, V., and Briand, L. “How to Reuse Influences Productivity in Object-Oriented Systems,” *Communications of the ACM* (39:10), 1996, pp 104-127.

Bentler, P., and Bonnett, D. “Significance Tests and Goodness of Fit in the Analysis of Covariance Structures,” *Psychological Bulletin* (88), 1980, pp. 588-606.

Byrd, T. A., and Turner, D. E. “Measuring the Flexibility of Information Technology Infrastructure: Exploratory Analysis of a Construct,” *Journal of Management Information Sciences* (17:1), 2000, pp. 167-208.

Byrne, B. *A Primer of LISREL: Basic Applications and Programming for Confirmatory Factor Models*, Springer-Verlag, New York, 1989.

Cooprider, J., and Henderson, J. “A Multi-Dimensional Approach to Performance Evaluation for I/S Development,” Working Paper No. 197, Center for Information Systems Research, Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA, 1989.

Davenport, T. *Process Innovation: Reengineering Work Through Information Technology*, Harvard Business School Press, Boston, MA, 1992.

DeLone, W. H., and McLean, E. R. “Information System Success: The Quest for the Dependent Variable,” *Information Systems Research* (3:1), March 1992, pp. 60-95.

Fayad, M. E., Hamu, D. S., and Brugali, D. “Enterprise Frameworks Characteristics, Criteria, and Challenges,” *Communications of the ACM* (43:10), 2000, pp. 39-46.

Gerwin, D. “Manufacturing Flexibility: A Strategic Perspective,” *Management Science* (39:4), 1993, pp. 395-410.

Gibson, V., and Senn, J. “System Structure and Software Maintenance Performance,” *Communications of the ACM* (32:3), 1989, pp. 347-358.

Glass, G., and Hopkins, K. *Statistical Methods in Education and Psychology*, Allyn and Bacon, Needham Heights, MA, 1984.

Guinan, P, Cooprider, J., and Faraj, S. “Enabling Software Development Team Performance During Requirements Definition: A Behavioral Versus Technical Approach,” *Information Systems Research* (9:2), 1998, pp. 101-125.

Hamilton, S., and Chervany, H. “Evaluating Information System Effectiveness, Part 1. Comparing Evaluation Approaches,” *MIS Quarterly* (5:3), 1981, pp. 55-69.

Harrington, H. James. *Business Process Improvement*, McGraw-Hill, Inc., New York, 1991.

Huber, G., and McDaniel, R. “The Decision-Making Paradigm of Organizational Design,” *Management Science* (32:5), 1986, pp. 572-589.

Humphrey, W. *A Discipline of Software Engineering*, Addison Wesley Longman, Reading, MA, 1995.

Hyman, R. "Creative Chaos in High Performing Teams: An Experience Report," *Communications of the ACM* (36:10), October 1993, pp. 56-68.

Joreskog, K. "Analyzing Psychological Data by Structural Analysis of Covariance Matrices," in *Contemporary Developments in Mathematical Psychology* (Volume 11), D. Icrantz (ed.), W. H. Freeman, San Francisco, CA, 1974, pp. 1-56.

King, W., and Epstein, B. "Assessing Information System Value," *Decision Sciences* (14:1), 1983, pp. 34-45.

Lord, F. M., and Novick, M. E. *Statistical Theories of Mental Test Scores*, Addison-Wesley Publishing Company, Wokingham, England, 1968.

Markus, M. "Paradigm Shifts-E-business and Business/Systems Integration," *Communications of the AIS* (4:10), 2000, pp. 1-45.

Miller, J., and Doyle, B. "Measuring Effectiveness of Computer Based Information Systems in the Financial Services Sector," *MIS Quarterly* (11:1), 1987, pp. 107-124.

Nelson, K. M., and Ghods, M. "Evaluating the Contributions of a Structured Software Development and Maintenance Methodology," *Information Technology and Management*, Winter 1998.

Paulk, M., Weber, C., Curtis, B., and Chrissis. M. "The Capability Maturity Model, Version 1.1," *IEEE Software*, July 1993, pp. 18-27.

Rossi, P., Wright, J., and Anderson, A. *Handbook of Research Methods*, Academic Press Inc., Orlando, FL, 1983.

Sommerville, I. *Software Engineering*, Addison-Wesley Publishing Company, Wokingham, England, 1989.

Thompson, J. *Organizations in Action*, McGraw Hill, Inc., New York, 1967.

Upton, D. "The Management of Manufacturing Flexibility," *California Management Review* (36:2), 1994, pp. 72-89.

Upton, D. "What Makes Factories Really Flexible? " *Harvard Business Review* (73:4), 1995, pp. 74-86.

Venkatraman, N. "The Concept of Fit in Strategy Research: Toward Verbal and Statistical Correspondence," *The Academy of Management Review* (14:3), 1989, pp. 423-444.

Yourdon, E. *Managing the Structured Techniques: Strategies for Software Development in the 1990's*, Yourdan Press/Prentice Hall, New York, 1986.