December 2003

The Role of Project Methodology in Large-Scale IS Project Management

J. Art Gowan
University of North Carolina at Wilmington

Richard Mathieu
Saint Louis University

The Role of Project Methodology in Large-Scale IS Project Management

J. Art Gowan, Jr.
Cameron School of Business
University of North Carolina at Wilmington
gowanj@uncwil.edu

Richard G. Mathieu
John Cook School of Business
Saint Louis University
mathieur@slu.edu

Abstract

This research examines the role of a formal project methodology on the performance of a large-scale information system project. Our research builds on the model presented by Banker et al. [1998] and Kirsch [2000] that show a positive relationship between complexity/size and project management practices as it relates to software project performance The proposed model was tested using data collected by means of a survey of 449 Fortune 1000 executives responsible for managing the Year 2000 (Y2K) project in their firm. Using structural equation modeling techniques, our results conclude that information system projects that are larger in size and higher in technical complexity have a greater degree of reliance on a formal project methodology in order to achieve a high level of project performance with respect to meeting a target date.

Keywords: IS project management, software project management, IS project methodology, Y2K problem

Introduction

Project management practices have been examined extensively in the software project management literature and to a lesser degree in the broader information systems literature. Software project management is defined as "the application of formal and informal project management techniques, tools and heuristics (collectively called project management practices) which are used by the project manager to motivate and guide a team to carry out a software project within a given set of constraints" (Kirsch 2000). There is empirical research that shows a positive relationship between formal project management practices and the performance of a software project. Deephouse et al. (1995-1996) show a positive relationship between two project management practices (project planning and cross-functional teams) and project performance. However, it is important to note that they found no positive relationship between other practices (design reviews, prototyping, frequent user contact, and maintaining a stable environment) and project performance. Guinan at al. (1998) found that the type of formal 'technology enabler' (CASE, structured methods, etc.) played a small role in determining project team performance. However, the role of formal tools in software project management is not always straightforward. For example, Banker et al. (1998) looked at two software design and development practices (code generators and packaged software) and found that the use of code generators actually increased the project hours devoted to software enhancement.

The use of a project methodology is a project management practice deemed to be very important in the conduct of an information system project. Research by Kirsch (1997) and Necco et al. (1987) have viewed systems development methodologies as a mechanism of behavior control by articulating precise steps in successfully developing a system. Kirsch (1997) concludes that practitioners often view a methodology as rough guidelines and that these procedures are not always intended to be judiciously followed. A project methodology can be viewed as a set of precisely described procedures or processes for achieving a standard task (Hackathorn and Karimi 1988). In this context, Guinan et al. (1998) found that group production processes (in this case effective plans and procedures) are positively related to software development performance.

The purpose of this paper is to investigate the role of project methodology in large-scale information system project management. A review of the literature indicates that there is a lack of empirical investigation of the issues related to the characteristics associated with large-scale information system (IS) projects, the formal use of a project methodology within an IS project, and the performance of the project. The unique characteristics of Y2K projects (the distributed nature of the computer assets for testing, the time criticality of the project, and the large demand for administrative resources) make it good laboratory to reexamine and broaden our understanding the factors related to IS project performance (Zmud and Kappelman 1997). This paper investigates the relationship between two IS project characteristics (technical complexity and project size), the use of a project methodology, and project performance (target date for project completion). We direct our research to answer the following question: what project characteristics influence the use of a formal project methodology in successful large-scale IS projects?

Prior research suggests that project characteristics directly affect project management practices. Zmud (1980) states that technological complexity, the degree of novelty of the application, technological change and project size influence the outcome of large software projects. Kirsch (2000) states the importance of understanding the role of complexity in software project management and concludes that 'exactly how project management practices will vary as a function of project complexity is a question for empirical research to examine'. The definition of project size and the complexity of a project has, at times, been blurred in the literature. In this research, the relationship between software complexity (Banker et al. 1998; Banker et al. 1993), task complexity (Kirsch 1996), and project size (Banker and Slaughter 1997) to project management practices is examined. Kirsch (2000) states that complexity is often thought of in terms of project size, and that large software project consists of many interrelated parts, while others (Banker et al., 1998) have made a distinction between the complexity of the technology and the size of the project. In this research we make a clear distinction between technical complexity and project size.

Research Methodology

The proposed model was tested using data collected by means of a survey sent to 2,773 IT professionals responsible for managing the Year 2000 (Y2K) project in their firm. A total of 449 usable surveys were returned in the first quarter of 1998. An introduction to the survey clearly explained that the "following questions address the distributed enterprise" with respect to their organization's Y2K compliance effort, and defined the distributed enterprise as "including PCs, servers, and remote devices, such as palmtops and laptops". The survey respondents represent a broad range of Fortune 1000-level companies, government agencies, and educators. Respondents were senior-level IT professionals. The majority of organizations responding (81%) have more than 5000 employees, with 8% employing more than 50,000. The annual revenue of 77% of the respondents is greater than $500 million, with 15% indicating an annual revenue exceeding $5 billion.

Project Size was developed as a latent variable based upon three items identified as "Annual Revenue", "Number of Employees", "Number of Workstations", and "Geographic Reach". These are not meant to be an exhaustive list of all the dimensions of project size, but are similar to other measures previously used in this field of study based upon the review of the literature. Technical Complexity was measured using three distinct survey items identified as "Rate of Software Change", "Non-Compliant Software", "Non-Supported Software", and "Non-Compliant Hardware". The first three items relate to software and the last to hardware. Each was analyzed separately as manifest variables. Project Methodology was developed as a latent variable based upon a survey item with five sub-items, each reporting the status of each stage of a typical Y2K compliance methodology. Each stage was described as either, "Yet to be considered", "Planned" or "Implemented" with respect to the project timeline. An information systems methodology should be tailored to the specific project. In this research, all participants were engaged in Y2K remediation projects. Thus we were able to inquire about specific methodological components specific to the Y2K project. We looked at five components to the methodology: (1) problem identification, (2) risk assessment, (3) cost calculations, (4) compliance planning, and (5) testing and verification. These five specific methodology components were derived from the literature on software project management, enterprise system implementation and Y2K project management. Target Date was used as a measure of project performance. The respondents were asked to report which half of which year was the projected target date for completion of the Y2K project, as of the time of the survey. A response of "Information not available" was treated as missing for purposes of analysis. Almost 95% of the respondents provided a response which then allowed it to be included in the analysis.

The two latent variables developed as Project Size and Project Methodology constructs, were assessed for reliability by calculating Chronbach's alpha. Chronbach's alpha for the Project Size construct was 0.75 and 0.86 for the Project Methodology construct, which fall within an acceptable range (Nunnally, 1967, p.226). Convergent and discriminant validity of the purported constructs were assessed by performing principal component factor analysis using varimax rotation. All indicators loaded onto their proposed factor, falling within an acceptable range (Nunnally, 1967, p.369), indicating both convergent and discriminant validity, shown in Table 1 below.

Table 1. Confirmatory Factor Analysis

Confirmatory Factor Analysis of Principal Components		
	Component	
	Project Size	Methodology
Annual Revenue	.832	-.080
Number of Employees	.848	-.060
Number of Workstations	.806	.148
Geographic Reach	.502	-.146
Problem Identification	-.180	.706
Risk Assessment	-.143	.807
Risk Correction	-.123	.840
Compliance Planning	. 053	.831
Risk Management	-.018	.768

Results

In order to test the proposed, structural equation modeling (SEM) was used. This form of analysis has been used with greater frequency in the IS field in the past few years and is an example of a second generation of multivariate analysis (Chin, 1988). Path analysis was performed using AMOS, Analysis of Moment Structures, (Arbuckle and Wothke 1999). An initial model was tested as a recursive, saturated model resulting in all possible paths tested in an exploratory mode. A number of path coefficients were found to be insignificant. In addition, the Technical Complexity measure of "Non-Compliant Hardware" and "Rate of Software Change" had no statistically significant relationship with Methodology or Target Date.

A revised model was analyzed with all insignificant paths eliminated. The final revised path model is shown in Figure 1. The results from the model are shown in Table 2. Assessment of the revised model resulted in a chi-square = 178.58 (df = 53, p=0.00), CFI = .990, NFI = .986, RFI = .980, and RMSEA = .064. All remaining path coefficients were found to be statistically significant. Table 2 provides the standardized path coefficients and critical ratios for each.

The measure of project size was significantly related to the intervening project methodology indicator, which was in turn significantly related to project performance. This indicates that as the size of an IS project grows increasing usage of a project methodology are found in IS projects which better meet their target date. Two of the original technical complexity manifest variables were to a lesser degree (<.10) significantly related to the intervening project methodology indicator. This indicates that as some dimensions of technical complexity (non-compliant software and non-supported software) increase there is increased use of a formal project methodology.

Table 2. Results for Final Path Model

Hypothesis	Path	Hypothesized Relationship	Path Coefficient	Hypothesis Supported – Critical Ratio?
1	Project Methodology → Project Target Date	+	.21	Yes (4.25)**
2	Non-Compliant Software → Project Methodology	+	.08	Yes (1.73)*
5	Non-Supported Software → Project Methodology	+	.09	Yes (1.81)*
7	Project Size → Project Methodology	+	.20	Yes (3.74)**

** significant at (p=.01)
* significant at (p=.10)

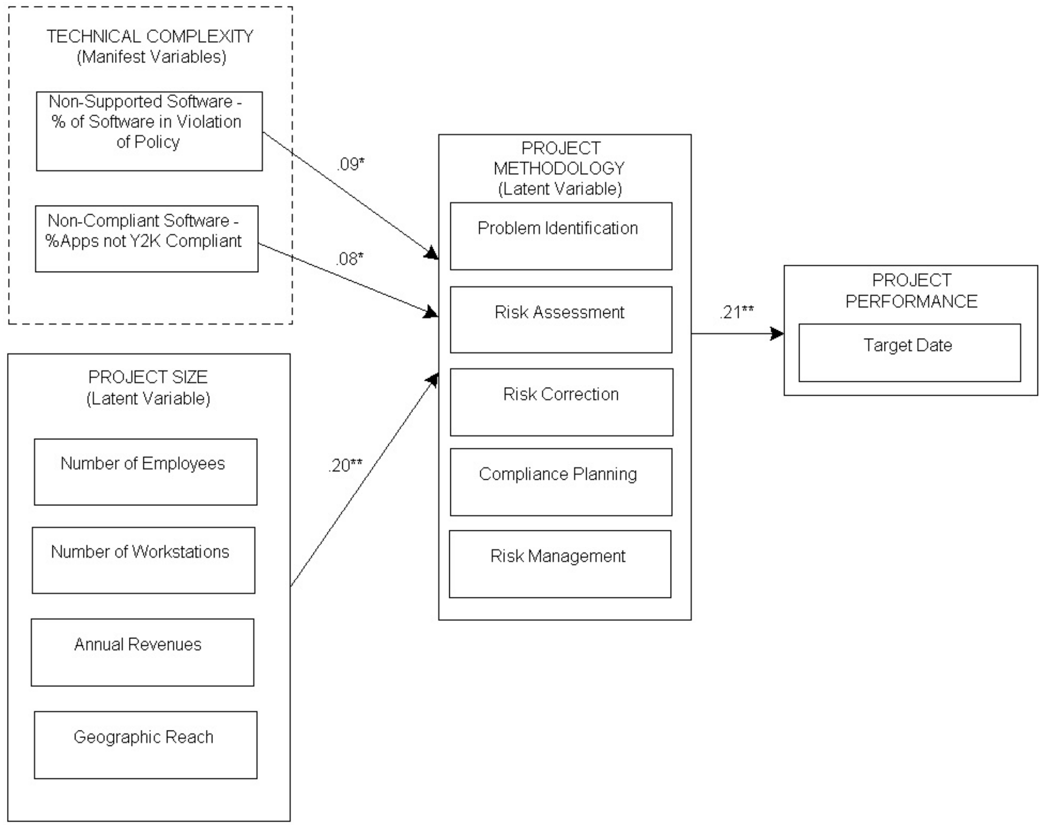

Figure 1. Final Path Model

Discussion and Conclusions

Our study has limitations that must be noted. First, a response rate of approximately 16% may have introduced bias, but the respondents did represent a broad range of Fortune 1000 companies based upon a number of demographic survey items. The only project performance measure of this study, target date, was self-reported, which can always introduce bias. Future research should consider additional measures of performance such as cost, satisfaction and business value. The only hardware-related indicator of technical complexity was ultimately dropped and was not included in the final model. Therefore, no specific conclusions could be drawn regarding hardware-related complexity. As the amount of hardware grows, including that which supports the basic architecture of a system (e.g. servers, and communications components), hardware complexity metrics may need to be given additional consideration. There are also a wide variety of other project characteristics that should be considered as well in this line of research. While the Y2K project did provide some degree of control for differences since the primary issues were the same for most organizations, it could also provide some limitations with regards to generalizability to other large-scale, enterprise-wide projects. For example, most Y2K issues did not involve the latest technologies, which had been designed to be Y2K compliant, for the most part. New issues will no doubt arise with the implementation of large-scale new technologies.

The goal of this study was to examine the role of a formal project management methodology on large-scale IS projects. We implemented the research using a dependent measure of IS project performance based upon a self-reported measure of target date (the expected date by which the Y2K project would be completed). The initial model analyzed included three software complexity measures and one hardware complexity measure. Two measures of complexity (non-compliant hardware and rate of software change) were not found to be significantly related with project methodology or project performance. This was specifically one of the most important hardware components that was of greatest concern with respect to Y2K compliance.

The message to IS project managers should be clear: As IS projects grow in size, and technical complexity, a formal project specific methodology will more than likely be required. Large-scale, enterprise-wide projects such as ERP implementations, webbased eCommerce systems that cut through the supply chain, and enterprise network management systems will be faced with issues similar to those addressed in the Y2K project. Hardware technical complexities may have more impact than those found in this study due to the distributed enterprise continuously expanding to include more and different types of platforms and system environments. Regardless of they type of complexities, or the fact that more IS managers are faced with larger IS projects, successful project performance will become more dependent upon the use of effective project methodologies.

References

Arbuckle, J. L. and Werner W. AOMS 4.0 User's Guide, Chicago, IL: SPSS Inc. and SmallWaters Corporation, 1999.

Banker, R. D., Davis, G. B., S., Slaughter, S.A. "Software development practices, software complexity, and software maintenance performance: a field study," Management Science (44:4), 1998, pp. 433-450.

Banker, R. D., Slaughter, S.A. "A field study of scale economies in software maintenance," Management Science, (43:12), 1997, pp. 1709-1725.

Banker, R.D., Datar, S., Kremerer, C.F., and Zweig D. "Software complexity and software maintenance costs," Communications of the ACM 36(11), 1993, pp. 81-94.

Chin, W. "Issues and opinion on structural equation modeling," MIS Quarterly (22:1), 1988, pp. vii-xvi.

Deephouse, C., Mukhopadhyay, T., Goldenson, D.R., Keller, M.I. "Software processes and project performance," Journal of Management Information Systems (12:3), 1995-96, pp. 187-205.

Guinan, P.J., Cooprider, J.G., Faraj S. "Enabling software development team performance during requirements definition: a behavioral versus technical approach," Information Systems Research, (9:2), 1998, pp. 101-125.

Hackathorn, R. D., Karimi, J. "A framework for comparing information engineering methods," MIS Quarterly (12:2), 1988, pp. 203-221.

Kirsch, L.J. "Software project management: an integrated perspective for an emerging paradigm," Zmud R.W. and Price, M.F. eds., Framing The Domains Of IT Management: Projecting the Future Through the Past*.* Pinnaflex Educational Resources, Cincinnati, OH, 2000, pp. 285-304.

Kirsch, L.J. "Portfolios of control modes and IS project management," Information Systems Research (8:3), 1997, pp. 215-239.

Kirsch, L.J. "The management of complex tasks in organizations: controlling the systems development process," Organization Science (7:1), 1996, pp.1-21.

Necco, C.R., Gordon, C.L., Tsai, N.W. "Systems analysis and design: current practices", MIS Quarterly*,* (11:4), 1987, pp. 461-475.

Nunnally, J. C. Psychometric Theory, McGraw-Hill, New York, 1978.

Zmud, R.W. "Management of large software development efforts," MIS Quarterly (4:2), 1980, pp. 45-55.

Zmud, R.W., Kappelman, L.A. "The year 2000 problem: a laboratory for MIS research (editor's comments)," MIS Quarterly (21:4), 1997, pp. iii-iv.