A Study on the Dimension of Quality Metrics for Information Systems Development and Success : An Application of Information Processing Theory+

Joon M. An *

〈Abstract〉

Information systems quality engineering is one of the most problematic areas in practice and research, and needs cooperative efforts between practice and theory [Glass, 1996]. A model for evaluating the quality of system development process and ensuing success is proposed based on information processing theory of project unit design. A nomological net among a set of quality variables is identified from prior research in the areas of organization science, software engineering, and management information systems. More specifically, system development success was modelled as a function of project complexity, system development modelling environment, user participation, project unit structure, resource availability, and the level of iterative nature of development methodology.

Based on the model developed from the information processing theory of project unit design in organization science, appropriate quality metrics for each variable in the proposed model are matched. In this way, a framework of relevant systems development and success quality metrics for controlling systems development processes and ensuing success is proposed. The causal relationships among the constructs in the proposed model are proposed as future empirical research for academicians and as managerial tools for quality managers. The framework and propositions help quality manager to select more parsimonious quality metrics for controlling information systems development processes and project success in an integrated way. Also this model can be utilized for evaluating software quality assurance programmes, which are developed and marketed by many vendors.

+ This research has been supported by New Faculty Research Fund of Konkuk University

* Department of Management Information Systems, School of Business administration, Konkuk University

1. Introduction

System failure is a common phenomenon rather than an unusual one. Problems lie not in the lack of methods, tools and managerial control, but rather in their fragmentation. Without integrated approaches to systems development quality management, the dynamic interplay involving people, methodology, and tools cannot be captured or understood [Chen and Nunamaker, 1989]. Information systems and software engineering researchers have focused on each component of systems development processes, activities, and quality metrics separately. Software engineering research community has concentrated on rather narrower concept of software metrics. In practice, software engineers or project managers usually collected and analysed software metrics to do input/output analysis without any causal understanding on related software metrics.

A wide range of systems quality management tools or software metrics such as Capacity Maturity Model (CMM) by Software Engineering Institute and ISO 9001 by ISO, are used to increase the productivity and efficacy of the systems development processes. But, very little is known concerning which information systems development projects are most conducive to the proper use of software quality metrics [Cerveny, Garrity, and Sanders, 1990], and how much and in what way the software quality metrics are interrelated to each other. There might be relationships among systems development quality metrics (development methodologies, project unit structure, user involvement, modelling environment, and resources) in a complementary fashion. The effectiveness of each internal metrics depends to some extent on the other metrics in a project. Prior research in the systems development quality engineering and systems development areas is typically not theory-driven. Furthermore, major systems development quality metrics[1)] such as the systems development methodology, software development tools, user participation, and system success have been evaluated in fragmented ways. Research into the system development process and success has suffered from the lack of a nomological net, i. e., the theoretical relationships among a set of constructs [Wrigley and Dexter, 1991].

The motivation of this research is to set up a theory-based systems development process and ensuing success quality (SDPSQ) model, a set of propositions and a framework of SDPSQ metrics. The research model views system

1) Systems development process and success quality metrics intrics include the metrics for measuring project complexity, systems development processes, and systems development success. They include software engineering quality metrics as a subset.

development project as a managerial and a controlling processes for dealing with the complexity of systems development project [Zmud, 1980]. Key systems quality management considerations, such as the selection of tools, project unit structure, and problem-solving strategy, are included in the model, in the guidance of information-processing theory of project-unit design. [Galbraith, 1977; Tushman and Nadler, 1978]. Based on the theoretically proposed SDPSQ model, a set of propositions and a framework of SDPSQ metrics are proposed.

Section two provides a review on major research in the areas of systems development and success. In Section three, information processing theory of project unit design is summarized to relate the constructs reviewed in the Section two to the constructs in the theory. In section four, systems development processes and project unit design strategies will be mapped to each other for formulating the SDPSQ model in Section five. Five propositions and a framework of SDPSQ metrics are detailed in Section six. Section seven summarizes conclusions and future research directions.

2. Literature Review on Information Systems Development Process and Systems Success

This section provides a review of major research streams in the system development process and success areas. It includes project complexity in systems development, the appropriate use of development tools and methodologies, the level of user involvement, the experience or knowledge level of related parties concerning a problem domain, the management structure of project-unit, and systems development success. The review provides a theoretical foundation for developing an integrated SDPSQ model based on the information processing theory of project unit design.

2.1 Complexity in systems development project

Most problems in systems development activities can be traced to the complexity inherent in software development environment. Research on system development and system success emphasizes the importance of the project complexity in selecting an appropriate methodology, development tool, project-units structure, and resources [Semprevivo, 1980; Davis, 1982]. But there has been little theoretical and empirical research effort to propose and validate the construct of project complexity.

An example of a system development project that has low complexity is one that contains a small number of simple task procedures which exhibit few exceptions and one where the input/output formats are simple, well defined, and unchanging. High complexity would be demonstrated by a system with many procedures, many exceptions to be processed, and input/output formats that are difficult to define and highly subject to many changes [Franz, 1985]. Franz [1985] defines complexity as the characteristic of developed systems.

Zmud [1980] observed that the problems associated with managing software development could be traced to the complexity that pervaded software development. This uncertainty emanates from numerous sources which include complex or state-of-the-art technologies being implemented, the influence of technological or environmental change, and the physical size of a software product. The complexity of an information system development project is considered to be a function of its size, structure, and technologies implemented for constructing the information system. McFarlan [1981] suggests that size, structure, and applied technologies used to develop a system effect project risk.

Brooks [1987] pointed out two sources of complexity in systems development environments. One relates to the inherent complexity of the problem being solved and the other is related to tools, languages, and design approaches.

Iivari and Koskela [1987] observed two main sources of the complexity: the scope of application domain (e. g. the number of transaction types processed) and the inherent multidimensionality of information systems. They noted that problems of system development were caused by the complexity of the system s being developed and the fuzziness of information requirements. They define system complexity as the composite of software complexity and information requirements stability.

Tait and Vessey [1988] performed a field study to see the relationship between system failure and systems complexity, and the mediation effect of user involvement to decrease the systems complexity. They found that systems complexity was directly related to system failure.

In sum, the complexity of systems development project is a major contingency factor influencing ensuing systems development processes and systems development success. Even though software engineering research has developed some software metrics such as LOC (lines-of-codes) and function points to measure the project complexity, those metrics should be supplemented with the broader concepts of project newness, analyzability, and variability, which have been proposed and empirically supported in organization science and management information systems research.

2.2 Systems development tools and methodologies

Information systems researchers identified development tools and methods as possible causes for system success and failure. Given that information system development projects are purposive activities for developing information systems, the manifestations of modelling activities at the project level are an important control strategy for overcoming project complexity and acquiring quality information systems. There is a wide variety of software development tools and methods currently available for increasing the productivity and effectiveness of a software development project. The structured revolution has brought in a wide range of development tools and techniques.

The analyst has a wide range of modelling tools and techniques from which to select and applies them in a wide variety of ways to the modelling process. While early structured tools and methods concentrated on improving coding practice for facilitating the lower levels of abstraction, current structured tools and methods also assist in higher levels of abstraction. In essence the system development modelling processes can control project complexity at each level of abstraction by selecting appropriate tools and methods.

Kydd [1989], for example, suggested that the proper methodology and tool should be context dependant. The context included the degree of uncertainty and equivocality, the stage of the development cycle, and technical maturity of the organization. For example, prototyping and group meetings between users and analysts could be used to reduce excessively equivocal and uncertain environments.

Hackathorn and Karimi [1988] developed a framework for comparing information engineering methods based on two dimension: depth and breadth. After placing each tool in a specific dimension according to its characteristics, they argued for a link between conceptual planners and pragmatic developers. The point was that system development methodologies and tools should facilitate interaction among interested parties throughout the entire development process to decrease project complexity.

In a related line Yadav, Bravoco, and Chatfield [1988] proposed a framework evaluating tools according to syntax, semantics, communicating ability, and usability. The implication is that a systems analysis tool should facilitate communication as well as support the organizational modelling process.

Mantei and Teorey [1989] focused on behavioral issues that arose over the course of the development life cycle and proposed techniques for gathering a variety of human-oriented information. For example, in the initial phase of systems development process, focus groups might be used to encourage users to elaborate on their problem domain.

On the empirical side of the research, Card, Garry, and Page [1987] reported that no particular systems development technique increased productivity significantly. The use of systems development techniques considered in the study produced approximately a 30 percent increase only in software reliability. They concluded that the effectiveness of any technique and the importance of specific non-technology factors, including programmer effectiveness, resources, data complexity, and the nature of the software application, could explain productivity at the project level.

Banker, Datar, and Kemerer [1991] reported that in maintenance projects, structured analysis and design tools did not contribute to the productivity of the project and suggested more empirical research in this area. To observe the impact of the structured analysis and design methodology, dummy variables indicating the use or absence of this technique were included in their model.

Cerveny and Joseph [1988] studied the software enhancement projects involving identical information requirements and they reported that firms using structured system design and programming techniques expended twice as much effort to implement the requirements as those using non-structured approaches.

In a comparison of traditional systems development life cycle approaches and prototyping methods, Mahmood [1987] found that the technique selected was related to project characteristics and the type of decision making the system was to support.

According to the previous studies, system specification tools and methods can be conceptualized as a way to control the complexity of systems development project. The functionalities of modelling tools suggested by previous studies are summarized in Table 1.

2.3 User involvement and participation

Current design philosophy suggests that more user involvement in information systems development is better. In fact, many structured techniques were employed to increase the involvement of users during system development processes. The common wisdom that user involvement may lead to successful system construction can actually be traced to concepts in the group psychology literature on interpersonal communication, group problem-solving, and motivation.

In a sense, the user involvement construct

Table 1: Review on functionalities of modelling tools

Author (year)	Functionalities
Kydd(1989)	Uncertainty reduction Eqivocality reduction
Hackathon(1988)	Depth: conceptual vs. pragmatic Breadth: planning vs. development
Yadav et al.(1988)	Syntactic Semantic Communicability Usability
Mantei (1989)	Gathering a variety of human oriented information
Mahmood(1987)	Adaptability to project and decision characteristics User/designer satisfaction
Adelson (1985)	Flexibility to designer's experience and expertise
Henderson et al. (1990)	Production technology: representation, analysis and transformation Coordination technology: control and cooperation Organization technology: support and infrastructure
Jarke(1986)	Reconciliation of divergent information among experts Negotiation

is a surrogate variable for measuring the effectiveness of interpersonal communication and the group problem solving process [Garrity, 1988].

After examining the link between user involvement and indicators of system success, Ives and Olson [1984] concluded that current research was based on inadequate theory and methodology. They proposed a new conceptual model illustrating the relationship between user involvement and success. According to their conceptual model, the user involvement construct might be represented by cognitive factors and motivational factors, which determine the system quality and system acceptance.

After reviewing information systems research and reference disciplines, Barki and Hartwick [1989] argued for a separation of the constructs of user participation and user involvement. They defined user involvement as a subjective psychological state during the systems development process and user participation as a set of activities performed by users during system development process.

As a consistent factor of contributing to systems development success, the type and numbers of user involvement and participation through systems development project will be included in SDPSQ model and metrics.

2.4 Knowledge and experience level of project members

The knowledge and experience level of an analyst in a specific domain have been found to be an important resource for supporting systems design process [Adelson and Soloway, 1985]. Also, the effective utilization of a tool or methodology is contingent on the designer's experience with the object being designed and the domain context.

Vitalari [1985] studied the foundations of expertise in systems analysis. He identified five categories of expertise including: core system analysis domain knowledge, organization specific knowledge, high-rated knowledge, functional domain knowledge, and knowledge of techniques and methods [Vitalari, 1985]. He observed that effective analysts use analogical reasoning, goal setting, hypothesis management, and certain problem solving strategies.

What skills do users and analysts consider important in analysts? Green [1989] found that users consider technical skills an essential component of analysts skills, whereas analysts view interpersonal skills as a very key element of their job. Perhaps a better understanding of the respective roles of users and analysts in the development process can lead to an effective partnership between the two parties. Some researchers suggest that the analysts can occupy several archetypical roles during the development process. Hirschheim and Klein [1989] suggest the following roles: the analyst as facilitator, the analyst as a labor partisan, the analyst as a social emancipator, and the analyst as an expert. Systems designer's experience on the tools, methodologies used and domain specific knowledge have been found that they influence systems development productivity and final systems success. Systems development project managers can accumulate knowledge and experience metrics and relate those metrics to systems development success metrics to estimate the required skill level for future projects.

2.5 Project unit structure of systems development team

In situations where there is a team developing a project, the outcome is derived from multiple interactions among participating actors. The problems that result from the interaction between diverse team members have been studied from various points of view. From a sociological perspective, Borovits, Ellis, and Yeheskel [1990] suggested that the communication patterns and working procedures of the project team affected productivity and intragroup relationships during systems development. They recommend collaborative group work and multi-directional communication in situations involving complex tasks. Guinan and Bostrom [1986] found that the productivity of developers and users could be significantly increased by improving communication patterns. Salaway [1987] turned to organizational learning theory as a mechanism to enrich user/analyst communication. After surveying the attitudes of data processing and user personnel towards their roles in system development, Cronan and Means [1984] concluded that the ability of users and analysts to take advantage of one another's expertise was a function of communication. Improving communication patterns appears to be a consistent theme.

Characterizing the system development process as an organizational change process is a way to confront systems implementation problems. Weitzel and Graen [1989] have turned this concept around slightly and looked towards joint problem solving competence as a variable affecting project effectiveness. They further conclude that the roles assigned to the user and analyst, as a function of the development methodology, affect the relationship between analyst and user.

Conflict resolution has been posited as an important construct for explaining the effectiveness of the systems development process [Robey, Farrow, and Franz, 1989]. In fact conflict resolution, communication, and issues in user participation are intertwined.

The real problem lies in the structure of project team. Project team structure inhibits the efficient usage of project members as problem solvers and the opportunity for feedback, error correction, and the synthesis of different points of views. White [1984] offered preliminary evidence that the project unit structure affected project quality and he suggested that further research in terms of the quality metrics of project unit structure was needed.

2.6 Systems development success

The success of an information system should be evaluated from both user's and analyst's perspective. System development success from user's point of view is typically operationalized using perceptual measures such as user satisfaction, system effectiveness, and use of the system. Total hours for finishing a project [Banker, Datar, and Kemerer, 1991], perceived level of user satisfaction [Deutsch,1991], and perceived efforts to implement a project enhancement [Cerveny and Joseph, 1988], and software reliability [Card, Garry, and Page, 1987] are used for measuring systems success from an analyst's point of view. Considering that system success is a multi-faceted concept, two measures including user information satisfaction and project success are adopted for evaluating systems success quality. The measures for the efficiency of solution or project success can be implementation, software reliability, and maintenance cost. The measures for the effectiveness of solution can be user information satisfaction with the developed system.

2.7 Summary

In previous research, to explain system development success, dominant constructs such as the complexity of systems development project, functionalities of modelling tools, user involvement, project unit structure, and skill level as a major resource were introduced into a research model to be developed in the next section without considering the relationships among the constructs.

Even though some research models, which integrated relevant factors to system success, were proposed, those models were not developed in guidance of sound theories. If a systems development process is viewed as a managerial process to deal with project complexity, key managerial considerations, such as project management guidelines, selection of tools, project unit structure, and problem-solving methodology, should be evaluated in an integrated way in order to explain the systems development success. This view goes with the observation that researchers in management information systems field should include interrelated variables in a model with strong theory [Weill and Olson, 1989].

In addition, previous studies in management information systems have not included or refined the constructs or factors developed in software engineering quality research. To set up SDPSQ model, information processing theory of project unit design from organization science was adopted. Detailed constructs in the theory are discussed in the following section.

3. Information Processing Theory of Project Unit Design

In previous section, major research efforts to explain systems development contingency, processes, and success were reviewed to relate the constructs in a meaningful way. To set up systems development process and success quality model, in this section, the information processing theory of project unit design is summarized based on some theoretical works [Galbraith, 1977 and Tushman, 1978] and empirical studies [Daft and Macintosh, 1981; Kmetz, 1984].

According to an information processing theory of project unit design [Galbraith, 1977], as project complexity increases, a project unit must take organizational design action to deal with the complexity. Galbraith's [1977] model suggests a more operational framework for linking a number of organizational design features to the level of complexity or information processing requirements facing a project unit

Tushman and Nadler [1978] suggest an information processing model of project unit in which project unit can be created to deal with specific aspects of the project environment [Katz and Kahn, 1966; Lawrence and Lorsch, 1967; Thompson, 1967]. In the model, the components of the project complexity were split into three parts including project characteristics, project environment, and inter-unit project interdependence.

The project design actions to overcome the complexity constitute the strategic choices taken by the project unit. It can proceed in either one of two general ways. First, it can reduce the amount of information processing that is required to overcome complexity by environmental management, creation of slack resources, and creation of self-contained projects. Secondly, the organization can increase information processing capacity by investing in modelling tools and the creation of lateral relations.

In the following subsections, each construct in the information processing model, including project complexity and project design strategies, will be summarized for developing propositions and a framework of SDPSQ metrics.

3.1 Project complexity

Tushman [1979] defined the complexity based on the characteristics of a project as important determinants of complexity and of the information-processing requirements of project unit [Thompson, 1967; Weick, 1969; Duncan, 1973; Pennings, 1975]. The project-related complexity refers to attributes of the local environment confronting the focal unit or what Emery and Trist [1965] describe as the "causal texture" of project environment. The three sources of project-related complexity are project characteristics, project environment, and inter-unit project interdependence. Project characteristics are defined as the composite of project complexity and intra-unit project interdependence. The project environment refers to a static/dynamic dimension of a project. The inter-unit project interdependence is the degree to which a subunit is dependent upon other subunits in order to perform its project effectively.

If a cognitive or information-processing perspective on the project complexity is adopted, the complexity facing a project unit is a function of the number of information demands or stimuli per unit time and the extent to which judgements must be taken into account in processing the information [Driver and Streufert, 1969; Farace, Monge, and Russell, 1977].

This study adopts the complexity as an important dimension confronting the systems development project unit. The project complexity is composed of project newness, project analyzability, and project variability. This view is supported by theoretical and empirical research on system success, reviewed in section one.

3.2 Project design strategy

Project design strategy consists of

adjustment processes to deal with project complexity. That is, the project unit modifies its domain and relations with elements in its environment depending upon the complexity and costliness of the other strategies to deal with the complexity. Project unit can adopt one of or mix of the two general strategy: one is the strategy to reduce the need for information processing, and the other is the strategy to increase the capacity to process information. Those two strategies are detailed in the first column in Table 2.

A project unit can enter various cooperative schemes such as implicit cooperation, contracting, coopting, and coalescing to reduce the environment complexity. In that way a project unit can transfer all or part of the required amount of information processing to other project units.

The creation of slack resources reduces the amount of information that must be processed during project execution and prevents the overloading of a project unit. An example of using slack resources is the time-cost trade-off in project networks. Investing financial or human resources in a project can decrease the overloading of a project unit and increase the possibility of project success. The next method for reducing the amount of information processing is to change from a functional project unit to one in which each project unit has all the resources it needs to perform its project; that is, change the way a project is decomposed into subprojects. The strategy of self-containment shifts the basis from one based on input, resources, skill, or occupational categories to one based on output. It reduces the amount of information processing by decreasing the amount of information which should be exchanged for coordinating activities among members during project execution.

In yet another approach a project unit can invest in modelling tools which allow it to process information acquired during project execution without overloading the project unit. This strategy increases the information processing at the planning stage while reducing the number of exceptions or complexity which may overload the project unit in later stages. For example, by utilizing an information system, a project unit does not have to engage unnecessary efforts to process information created during project execution. Various man-machine combinations or information processing methods may increase the capacity of problem solvers in a project unit.

The last mode for coping with environmental complexity is to selectively employ lateral relationships which cut across a project unit. This mode moves problem solving activities or information processing processes to where the necessary knowledge exists. In this way it will decentralize problem solving activities without creating a self-contained project unit. For example, interdepartmental group problem solving can be a mechanism to reduce the information overload.

3.3 Application of information processing theory of project unit design

The information processing theory of project unit design was utilized to describe a complex workflow in aircraft electronics repair [Kmetz, 1984]. Daft and Macintosh [1981] did an exploratory test of the model on 24 project units to see that the complexity and the amount of information processed are closely related. The reported amount of information processing increased with both project variety and analyzability; the reported use of equivocal information decreased with project analyzability.

Daft and Lengel [1986] argued that complexity and equivocality were two forces that influence information processing in organizations. Depending upon the complexity composed of analyzability and equivocality, the type and amount of information required for project accomplishment are different. Daft and Lengel's [1986] framework was used to relate appropriate information development methodologies to the level of analyzability and equivocality of a project. Equivocality resolution during implementation processes was found critical to realizing improved productivity at the project level [Jones and Kydd, 1988].

The framework was also applied to understand design alternatives for organizing information systems activities in general [Zmud, 1984] and to suggest possible managerial strategies for large scale software development efforts [Zmud, 1980]. The two most direct strategies for achieving more effective management of software development are to reduce the absolute amount of complexity or complexity within a project and to facilitate information flow among problem-solvers confronted with complexity [Zmud, 1980].

Information systems development projects are very complex and require various design actions to process much information in short time of period. This information intensive environment is quite appropriate to apply the theory to integrate set of design actions detailed in section two.

4. Mapping of Project Design Strategies to Information Systems Development Strategies

Based on the above discussion on information processing theory of project unit design, the information systems development-unit is assumed to take design actions to realize information processing fit. Drazin and Van de Ven [1985] state that the project of researchers adopting the concept of fit is to identify the feasible set of processes and technology dimensions that are effective for different environments and to understand which pattern of process-technology linkage are internally effective.

According to Banker and Kauffman [1991], a large set of variables should be included in a more general model to evaluate the quality of software development activities. The variables suggested by them include: technical qualities of the tools, development team characteristics, organizational factors, and architectural factors. This study selects and integrates related constructs from information systems research according to the information processing theory of project unit design reviewed above. It seems reasonable to assume that for the highly information intensive activities such as system development projects, the limits of information processing capacity are frequently reached or

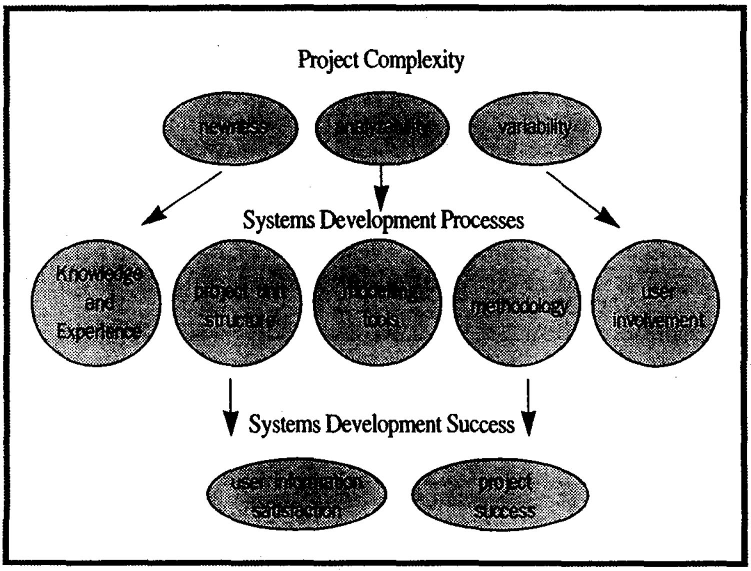
⟨Table 2⟩ Mapping of project design strategies to information systems development strategies

Project Design Strategy	Information Systems Development Strategy
[illegible]	
Environmental management: creation of cooperative scheme such as (1) contracting (2) coopting (3) coalescing	Contracting with outside vendor for a project (1) total outsourcing (2) contract development (3) partial contract development
Creation of slack resources: (1) extended completion date (2) use of slack resources (3) relaxing budget target	Investment in resources: (1) hardware/software (2) human resources (3) financial resources
Creation of self-contained task: (1) output based organization (2) reduced division of labor	(1) flexible structure of a project unit (2) efficient team structure
[illegible]	
Investment in vertical information systems: (1) formalization of problem solving symbol (2) increase of frequency of problem solving	(1) utilization of modelling tools and methods (2) iterative nature of development methodology
Creation of lateral relations (joint problem solving) (1) direct contact (2) liaison role (3) task force (4) team	(1) user involvement from functional department

exceeded by information processing requirements and that the difficulty in realizing information processing fit constantly constrains performance or success at the project level. The mapping between project unit design strategies and systems development strategies at the project level is shown in Table 2. The project design strategies will be split into two general strategies in this study: strategies to reduce the need for information processing and strategies to increase the capacity to process information. The strategies to reduce the need for information processing include contract with outside vendors, allocation of resources, project unit structure. The strategies to increase the capacity to process information include iterative nature of development methodology, utilization of modelling tools, and user involvement during the development process in a consistent and complete way. These two general strategies to deal with project complexity are assumed to affect system development success.

5. An Integrated Model: Systems Development Process and Success Quality (SDPSQ) Model

Based on the discussion in previous sections, in Figure 1, an integrated SDPSQ model is proposed. The model argues that the fit between project complexity and project design strategies at the project level determines system development success. Fit here is regarded as a mediating relationship [Venkatraman, 1989]: systems develop-ment strategies mediate the relationship between project complexity and systems development success. Specifically, project complexity is mediated by user involvement, development methodology, the system development modelling environment strategies, and resources for systems development success. The



<Figure1> Systems Development Process and Success Quality (SDPSQ) Model

unit of analysis for the proposed model is the information systems development project. Previous studies of the technology imperative paradigm suggested that the linkage between contingency and structure to predict performance at the project unit or project level was more meaningful than at the macro organizational level. Further, Cerveny and Sanders [1986] concluded that there had been little empirical support demonstrating the importance of macro organizational variables on system success. By matching related factors to systems development success at the project level, meaningful propositions and a framework of SDPSQ metrics can be proposed for future practical and empirical analysis. This model is unique in that it explicitly includes related constructs to systems development success in the guidance of a proven theory. By matching project complexity with the system development strategies, the net impact of each strategy in the model on systems development success can be assessed.

6. Propositions and A Framework of Systems Development Process and Success Quality Metrics

Based on the proposed model in Figure 1, the relationships among the constructs in the model are proposed in section 6.1. The propositions can guide systems engineering researchers to develope more detailed hypotheses for future theory development and empirical tests. This model also gives systems quality managers a frameworks for collecting a parsimonious set of quality metrics for future quality planning.

6.1 Propositions

The five propositions are derived from the information systems development process and success research detailed in section two and the information processing theory of project design discussed in section three.

Proposition 1: As the complexity of a project increases, the likelihood of the systems success decreases.

Proposition 2: As the complexity of a project increases, the extent of modelling activities positively correlates with systems development success.

Proposition 3: As the complexity of a project increases, the extent of user involvement activities positively correlates with systems development success.

Proposition 4: As the complexity of a project increases, the extent of knowledge and experience positively correlates with systems development success

Proposition 5: As the complexity of a project increases, the extent of the flexibility of project unit structure positively correlates with systems development success.

6.2 A framework of systems development process and success quality metrics

Based on the above propositions derived from research in organization science and management information systems, SDPSQ metrics developed from software engineering quality and information systems research can be organized as in Table 3. This framework is theory-based and practically meaningful in some respects. Previous research on software engineering has been focused on input and output metric analysis. In other words, project complexity metrics such as size of LOC (line of cords) or function points were utilized for predicting the budget or manpower for executing a software development project. The metrics suggested in Table three can supplement input/output metrics by supplying mediating metrics between the input and output metrics.

7. Conclusions Future Research Directions

Research projects aimed at developing a complex theory for organizing information systems activities are strongly advocated [Zmud, 1984]. Depending upon the complexity of an information development project, managerial arrangements and the system development modelling environment might have different effects on producing quality information systems.

This study proposes an integrated model to assess the interaction among the important factors which will affect systems success and incorporates systems development project complexity as an important factor influencing systems development processes and success. The complexity of systems development projects includes technological environment, project assignment, and reporting relationships [Baroudi and Ginzberg, 1986]. Each factor cannot be considered apart from others [Mahmood, 1987]. For example, user involvement should be evaluated with organizational arrangements such as project management strategy, resources commitment, and top management support [Kling, 1977].

Based on the integrated model of systems development process and success, a set of propositions and a framework of systems development and process quality metrics were proposed for future exploration. The set of propositions sets a starting point for theoretical expansion and empirical tests for academicians. The framework guides quality managers to collect more parsimonious set of interrelated quality metrics to optimize systems development process in a specific environment.

Table 3: Suggested systems quality metrics based on SDPSQ model

Constructs from information processing theory	Related variables from information systems research	Quality metrics
Project complexity	Project newness	(1) degree of project newness
	Project analyzability	(1) application components over project life cycle (2) programming complexity (3) data complexity (4) systems design complexity
	Project variability	(1) change over time (2) change request effort over time (3) module volatility over time
Strategies to reduce the need for information processing	Knowledge and experience	(1) effort (2) staff (3) computer resources (4) personnel skills (5) production support effort over time (6) systems engineering development productivity ratio (7) function points productivity ratio (8) documentation pages productivity ration (9) source statement productivity ratio
	Project unit structure	(1) project organizational complexity (2) type of project organization (3) reporting structure
Strategies to increase the capacity to process information	Modelling tools	(1) number of standards/ methodologies/ case tools for each stage (2) automated testing/ debugging/ code analysis/ (3) configuration management tools (4) level of documentation for modelling (5) number of tools and methods for each level of abstraction
	Development methodology	(1) documentation review session (2) iterative nature of applied modelling tools/ methodology
	User involvement and participation	(1) type and number of user involvement for each stage (2) type and number of user participation for each stage
Systems development success	User satisfaction	(1) service quality of information systems (2) number of user over time
	Project succes	(1) reliability (2) mean time to default (3) fault density (4) mean time to defect (5) mean time to failure (6) response time over time (7) down time over time

〈References〉

[1] Adelson, Beth and Soloway, Elliot, " The Role of Domain Experience in Software Design," *IEEE Transactions on Software Engineering*, SE-11, 11 (1985), 1351-1360.

[2] Amoroso, Donald L. and Cheney, Paul H., " Testing a causal model of end-user application effectiveness," *Journal of Management Information Systems*, Summer, 1991, Vol. 8, No. 1, 63-89.

[3] Banker, Rajiv D., Datar, S. M., and Kemerer, C. F.,"A Model to Evaluate Variables Impacting the Productivity of Software Maintenance Projects," *Management Science*, Vol. 37, No. 1, January 1991, 1-18.

[4] Banker, Rajiv D. and Kauffman, Robert J., "Reuse and Productivity in Integrated Computer-Aided Software Engineering: An Empirical Study," *MIS Quarterly*, 1991, September, 375-401.

[5] Barki, Henri and Hartwick, Jon, "Rethinking the Concept of User Involvement," *MIS Quarterly*, March 1989, 53-63.

[6] Baroudi, Jack and Ginzberg, Michael, "Impact of the Technological Environment on Programmer/ Analyst Job Outcome," *Communications of ACM*, June 1986, 546-555.

[7] Brooks, Fred P., "No Silver Bullet," *IEEE Computing*, Vol. 20, No. 4, April 1987, 10-19.

[8] Card, David N., Garry, F. E., and Page, G. T.,"Evaluating Software Engineering Technologies," *IEEE Transactions on Software Engineering*, Vol. SE-13, No. 7, July 1987, 845-851.

[9] Cerveny, Robert P., Garrity, E.J., and Sanders, G. Lawrence,"A Problem-solving Perspective on Systems Requirement," *Journal of Management Information Systems*, 6, 4 (1990), 103-122.

[10] Cerveny, Robert P. and Joseph, Daniel A., "A Study of the effects of Three Commonly Used Software Engineering Strategies on Software Enhancement Productivity," *Information and Management*, Vol. 14, 1988, 243-251.

[11] Cerveny, Robert P. and Sanders, Lawrence G., "Implementation and Structural Variables," *Information and Management*, 11, 1986, 191-198.

[12] Chen, Minder and Nunamaker, J. F., "Software Productivity: A Framework of Study and Approach to Reusable Components," *Proceedings of the Twenty Second Annual Conference on Systems Science*, Vol II, (January 1989), 959-968.

[13] Colter, M.A., "A Comparative Examination of Systems Analysis Techniques," *MIS Quarterly*, Vol. 8, No. 1, March 1984, 51-66.

[14] Cronan, Timothy P. and Means, Thomas L.,"System Development: An Empirical Study of User Communication," *Data Base* (Spring, 1984), 25-33.

[15] Daft, Richard L. and Lengel, Robert H., " Organizational Information Requirements, Media Richness and Structural Design," *Management Science*, Vol. 32, No. 5, May 1986, 554-571.

[16] Daft, Richard L. and Macintosh, Norman B., "A Tentative Exploration into the Amount and Equivocality of information Processing in Organizational Work Units," *Administrative Science Quarterly*, 26(1981), 207-224.

[17] Davis, G. B., "Strategies for Information Requirement Determination," *IBM Systems Journal*, Vol. 21, No. 1, 1982, 4 - 30.

[18] DeLone, William H. and McLean, Ephraim R., "Information Systems Success: The Quest for The Dependent Variables," *Information Systems Research*, Vol. 3,No. 1, pp.60-94.

[19] Deephouse, Chris, Mukhopadhyay, Tridas, Goldenson, Dennis, and Kellner, Marc, "Software Processes and Project Performance," *Journal of Management Information Systems*, Winter 1995-96, Vol.12, No.3, pp.186-205.

[20] Deutsch, Michael S., "An Exploratory Analysis Relating the Software Project Management Process to Project Success," *IEEE Transactions on Engineering Management*, Vol. 38, No. 4, November 1991, 365-375.

[21] Drazin, Robert and Van de Ven, Andrew, "Alternative Forms of Fit in complexity Theory," *Administrative Science Quarterly*, 30 (1985), 514-539.

[22] Franz, Charles R., "User Leadership in the Systems Development Life Cycle: A complexity Model," *Journal of Management Information Systems*, Fall 1985, Vol. 2, No. 2, 5-25.

[23] Franz, Charles R. and Robey, Daniel, " Organizational Context, User Involvement, and the Usefulness of Information Systems," *Decision Sciences*, Vol. 17 (1986), 329-356.

[24] Galbraith, Jay R., "Organizational Design: An Information Processing View," *Interfaces*, Vol 4, No. 3, May, 1974, 28-37.

[25] Galbraith, Jay R., *Organization Design*, Addison-Wesley Publishing, 1977.

[26] Garrity, E.J., A *Field Study of Prototyping Strategies*, Unpublished Ph. D. Dissertation, State University of New York at Buffalo, NY, Feb. , 1988.

[27] Glass, Robert L., "The relationship between theory and practice in software engineering," *Communications of ACM*, Nov., 1996, Vol 39, No. 11, pp. 11-13.

[28] Green, Gray I., "Perceived Importance of System Analysts' Job Skills, Roles, and Non-Salary Incentives," *MIS Quarterly* (June, 1989), 114-133.

[29] Guinan, Patricia J. and Bostrom, Robert P.,"Development of Computer-based Information Systems: A Communication Framework," *Data Base* (Spring,1986), 3-16.

[30] Hackathon, R.D. and Karimi, J., "A Framework for Comparing Information Engineering Methods," *MIS Quarterly* (June, 1988) 203-220.

[31] Henderson, J. C. and Cooprider, J. G.,"Dimensions of I/S Planning and Design Aids; A Functional Model of CASE Technology," *Information Systems Research*, 1, 3 (1990) 227-254.

[32] Hirschheim, R. A., "Assessing Participative Systems Design: Some Conclusions from an Exploratory Study,"

Information and Management, December 1983, 317-328.

[33] Hirschheim, Rudy and Klein, Heinz, 'Four Paradigms of Information systems Development, *Communications of the ACM*, 32, 10 (1989), 1199-1216.

[34] Iivari, Juhani and Koskela, Erkki, "The PIOCO Model for Information Systems Design," *MIS Quarterly*, September 1987, 401-419.

[35] Ives, Black and Olson, Magagrethe H., "User Involvement and MIS success: a Review of Research," *Management Science*, 1, 5 (1984) 586-603.

[36] Jarke, Matthias,"Knowledge Sharing and Negotiation Support in Multiperson Decision Support System," *Decision Support Systems*, 2 (1986), 93-102.

[37] Jones, Louise H. and Kydd, Christine T., "An Information Processing Framework for Understanding Success and Failure of MIS Development Methodologies," *Information and Management*, 15(1988), 263-271.

[38] Kan, Stephen, *Metrics and Models in Software Quality Engineering*, Addison Wesley, Reading, Mass., USA, 1995.

[39] Katz, D., and Kahn, R., *The Social Psychology of Organizations*, New York, Wiely Co., 1966.

[40] Kling, R., " The Organizational Context of User Centered Software Designs," *MIS Quarterly*, 1, 4, (Dec., 1977), 41-52.

[41] Kmetz, John L., "An Information Processing Study of a Complex Workflow in Aircraft Electronics Repair," *Administrative Science Quarterly*, 29(1984), 255-280.

[42] Kydd, Christine T., "Understanding the Information Content in MIS Management Tools," *MIS Quarterly* (September, 1989), 277-290.

[43] Lawrence, Paul, and Jay Lorsch, *Organization and Environment*, 1967, Harvard University Press.

[44] Mahmood, Mo A., "System Development Methods - A Comparative Investigation," *MIS Quarterly* (September, 1987) 292-311.

[45] Mantei, Marilyn M. and Teorey, Toby J.,"Incorporating Behavioral Techniques into the Systems Development Life Cycle", *MIS Quarterly* (September, 1989), 256-273.

[46] Martin, James and McClure, Carma, *Diagramming Techniques for Analysts and Programmers*, Prentice-Hall, Inc., Englewood Cliffs, NJ., 1985.

[47] Robey, Daniel, Farrow, Dana L., and Franz, Charles R.,"Group Process and Conflict in System Development," *Management Science*, 35, 10 (1989), 1172-1191.

[48] Salaway, Gail,"An Organizational Learning Approach to Information Systems Development," *MIS Quarterly* (June, 1987), 244-264.

[49] Sanders, Lawrence G. and Courtney, James F., "A Field Study of Organizational Factors Influencing DSS Success," *MIS Quarterly*, March 1985, 77-93.

[50] Semprevivo, Phil, "Incorporating Data Dictionary/ Directory and Team Approaches into the Systems Development Process," *MIS Quarterly*, September, 1980, 1-15.

[51] Tait, Peter and Vessey, Iris, "The Effect of User Involvement on System Success: A complexity Approach," *MIS Quarterly*, March 1988, 91-108.

[52] Thompson, J. D., *Organization in Action*, McGraw-Hill, New York, 1967.

[53] Venkatraman, N., "The Concept of Fit in Strategy Research: Toward Verbal and Statistical Correspondence," *Academy of Management Review*, 1989, Vol. 14, No. 3, 423-444.

[54] Vessey, I., and Weber, R., "Structured Tool and Conditional Logic: An Empirical Investigation," *Communications* Of ACM, 29, 1, 1986, 48-57.

[55] Vitalari, Nicholas P., "Knowledge as a Basis for Expertise in Systems Analysis: An Empirical Study," *MIS Quarterly* (September, 1985), 221-241.

[56] Weill, Peter and Olson, Margrethe H.,"An Assessment of the complexity Theory of Management Information Systems," *Journal of Management Information Systems*, Summer 1989, Vol. 6, No. 1, 59-85.

[57] Weitzel, John R. and Graen, George B.,"System Development Project Effectiveness: Problem-Solving Competence as moderator Variable," *Decision Science*, 20 (1989), 507-531.

[58] White, Kathy B.,"MIS Project Teams: An Investigation of Cognitive Style Implication," *MIS Quarterly* (June, 1984) 95-101.

[59] Wrigley, Clive D. and Dexter, Albert S., "A Model for Measuring Information Systems Size," *MIS Quarterly*, June, 1991, 245-257.

[60] Yadav, Surya B., Bravoco, Ralph R., Chatfield, Akemi T.,and Rajkumar, T. M., "Comparison of Analysis Techniques for Information Requirement Determination," *Communications of the ACM*, 31, 9 (1988) 1090-1097.

[61] Zahedi, Fatemeh, *Quality Information Systems*, Boyd and Fraser,Danvers, Mass., USA 1995

[62] Zmud, Robert W., "Management of Large Software Development Efforts, " *MIS Quarterly*, June 1980, 45-56.

[63] Zmud, Robert W.," Design Alternatives for Organizing Information Systems Activities, " *MIS Quarterly*, June 1984, 79-93.

[64] Zmud, Robert W., Mclaughlin, Curtis P., and Might Robert J., "An Empirical Analysis of Project Management Technique Implementation Success, " *Management Science Implementation*, Supplement 1, 107-132, 1984, JAI Press Inc.