1987

MANAGING THE USER RELATIONSHIP IM INFORMATION SYSTEMS DEVELOPMENT PROJECTS: A TRANSACTION GOVERNANCE APPROACH

Cynthia Mathis Beath
University of Minnesota

MANAGING THE USER RELATIONSHIP IN INFORMATION SYSTEMS DEVELOPMENT PROJECTS: A TRANSACTION GOVERNANCE APPROACH

Cynthia Mathis Beath
Carlson School of Management
University of Minnesota

ABSTRACT

This paper compares the effectiveness of two mechanisms for governing the relationship between an information systems development team and the new system's users. This relationship is traditionally governed using phased commitments and user involvement. Drawing on the organizational economics literature, the paper proposes a new view in which a project is characterized as a transaction, or an exchange, between IS and the users. Two alternatives for governing this exchange, one based on explicit, classical contracting and the other relying more on implicit, social contracting, are hypothesized to be differentially effective in governing exchanges of low or high difficulty, respectively. The model is explored in a field study at a single site and found to be supported, indicating that more rigorous tests of the model are warranted.

INTRODUCTION

Information systems (IS) development projects are a way of life in MIS departments. On these projects, project teams composed principally of IS specialists specify, develop (or obtain) and install an application system for business users in other parts of the same organization. There are many stresses and strains in systems work, but the relationship between the IS specialists and their organizational colleagues is one of the most problematic. Boehm (1981) notes that difficulties in this relationship can double the costs of a project. Since it is the users who will ultimately pass judgement on the system, and by extension, on the IS function, it would seem desirable to have a more satisfactory model for understanding, and thus managing, the relationship between IS project teams and their user community. This paper proposes and explores such a model.

The major business users of information systems include clerical personnel, first-level managers, staff specialists, and higher management (Davis 1974). In the IS literature the term "user" frequently goes undefined, but it usually includes that person or persons who (a) hold or control the resources being exchanged for the system (hence the term "client" or "customer"), and (b) will provide or use data from the new system. In theory and in practice this multitude is usually condensed to a "virtual user" who combines some of each characteristic; that is, he or she is high enough in the organization to influence the flow of resources and is also a knowledgeable participant in the business function to be supported. The term "user" is employed here in the same sense.

In IS development projects, two things need to be accomplished in the project/user relationship. First, to invest organizational resources wisely, the costs and features of the new system should be agreed upon in advance. Second, the user must find value in the installed system. That is, it is not enough that the IS department delivers a system that matches the specifications. It is also necessary that the system be useful. The two principal mechanisms by which we seek to accomplish these goals are aspects of the systems development life cycle (SDLC) and user involvement.

An SDLC has many features which support the day-to-day activities of the project team. Its primary support for investment management, however, lies in a series of checkpoints at which proposed costs and features of the system are reviewed by the users, providing them an opportunity to reconsider the decision to invest in the system. Beginning with the initial project authorization, the SDLC takes the

user and team through a series of agreements. Increasingly specific documents detail the agreed upon costs and features of the new system. Change control mechanisms record addenda to these agreements. Preference for this gradual and public commitment to IS investments is strong in today's organizations. (The principal difficulty with prototyping is that it threatens to violate this norm, and so prototyping is now seen as an alternative method of requirements elicitation (Zmud 1980) to be carried out within an SDLC-type framework of gradual commitment.)

User involvement is the other main mechanism used by project teams to manage the user relationship. While the efficacy of user involvement has been difficult to demonstrate (Ives and Olson 1984), it is believed to result in more accurate user requirements (Powers and Dickson 1973) and to increase the likelihood that the user will value the system (Swanson 1974; Markus 1983). User involvement is thus an important supporting mechanism for the agreements produced by the SDLC. That is, user involvement both improves the quality of the agreements produced by the SDLC and strengthens users' commitment to those agreements, increasing the likelihood that the user will use and favorably evaluate the IT investment.

However, user involvement is difficult to implement. Users have their own jobs to do, and they are not specialists. User involvement, particularly by those who are not clients, can muddle rational investment control. Who should make which decision? How can user decisions be made visible? Who bears the consequences of user-initiated delays or overruns? Difficulty in resolving these issues discourages user involvement.

The IS literature acknowledges that gradual commitment, based on specification of requirements, is difficult in the face of complexity, uncertainty, lack of structure, time pressure, and so forth (see Beath 1983). A variety of alternative mechanisms are suggested, but they have little theoretical basis and have not been widely accepted. This paper draws from the organizational economics literature for some advice on what to do about those more difficult projects. A model is proposed in which a traditional SDLC-based approach is appropriate for governing the project/user relationship in straightforward projects, as expected. More challenging projects, however, require a governance approach based on social contracting, which rests on a foundation of user involvement.

A NEW PERSPECTIVE

The organizational economics literature (Barney and Ouchi 1986) considers the question of why organizations take one form or another. It focuses on transactions, or exchanges of resources, and asks, "Why are some exchanges carried out within the organization and others outside?" The opinion of this literature is that an exchange will require more elaborate governance mechanisms if the exchange triggers either or both of two human weaknesses--bounded rationality (small minds) and self-interest (hard hearts). Markets are considered to use the least elaborate governance mechanism (using a combination of prices, classic contracts and competition), and clans the most elaborate (based on traditions, social agreements and a common world view) (Ouchi 1980; Williamson 1975; Williamson and Ouchi 1981). More elaborate strategies, while more powerful, are more costly to establish and maintain. Thus, to minimize governance costs, governance mechanisms should be matched to the uncertainty and strategic opportunities in the exchange, which, if ungoverned, will cause the exchange to fail or to be concluded inefficiently.

Why is this literature important in managing the project/user relationship? First, consider that an internal systems development project implies an exchange between IS and its user community. That exchange must be governed to minimize the effects of bounded rationality and self-interest. This is accomplished by exchanging information about the costs and features of the system often enough to monitor the correctness of the investment decision. If the project is a simple one, this information exchange can be concluded quite easily. If the project is complex, more elaborate governance mechanisms are needed. Compared to the traditional view of governing the project/user relationship, shown in Figure 1, this new view, summarized in Figure 2, is not radically different but does include some interesting additional implications.

The sources of project difficulty in Figure 2 include not only the usual uncertainty and complexity issues (small minds issues, for the most part), but also considerations of previous and future exchanges (hard hearts issues). Previous investments relevant to the current exchange -- such as IS training in a particular technology -- or ignorance of future

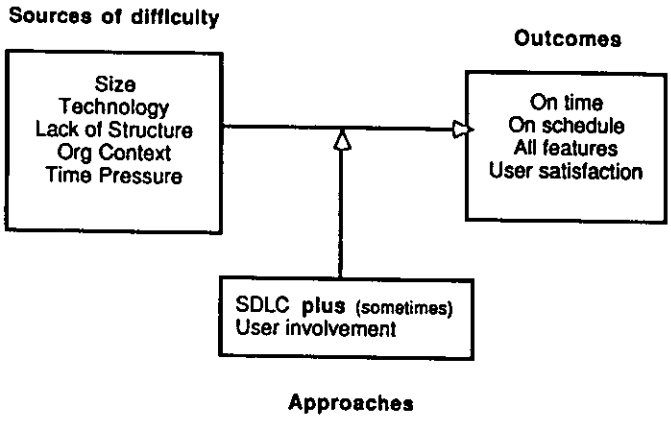

Figure 1. How to Manage the Project/User Relationship: The Traditional View

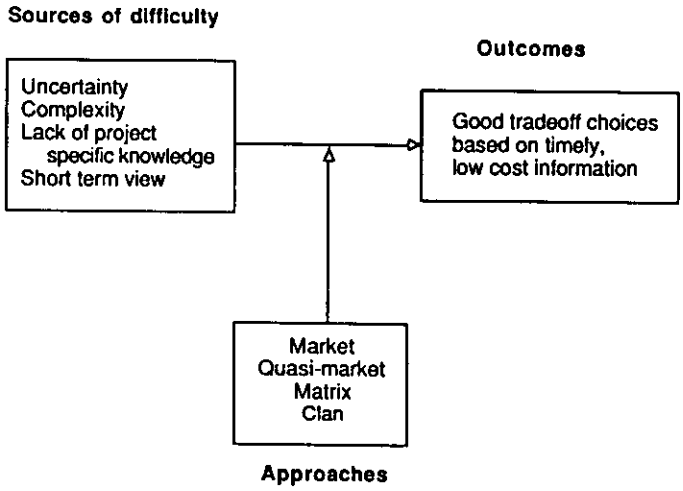

Figure 2. How to Manage the Project/User Relationship: The Transaction Governance View

payouts from the current exchange make design choices, and thus project investment decisions, more problematic. For one thing, prior investment may activate self-interest by creating sunk costs or small numbers bargaining situations. For another, a short term view is a disincentive for investing in more elaborate governance.

The list of potential project/user governance mechanisms could be quite lengthy, as any approach equal to the "small minds" and "hard hearts" problems of a particular project would qualify. Market governance would be appropriate when purchasing software from vendors. Clan governance is perhaps appropriate for end-user computing. The SDLC approach to exchange governance discussed above is equivalent to *quasi-market* governance. Quasi-market governance is basically an internal market, in which prices and contracts are used to control small minds problems and the employment relation controls hard heart problems. It includes several classic contract notions: prespecification of acceptance criteria, infrequent checkpoints, measurement of progress and costs by comparison to written, explicit agreements. Another possible governance approach is *matrix* governance (Beath 1983). Matrix governance is based on social contracting notions, in which the small minds problems are dealt with in implicit, weakly specified but mutually understood agreements, and the hard hearts problems are dealt with by a sense of personal obligation and commitment. Matrix governance involves shared responsibilities, more frequent checkpoints and contact with users, and social assessment of progress and outcomes, which may continue beyond system installation.

In the new view, the objective of a project/user governance mechanism broadens from "delivering all features on time and on budget" to "providing information regarding the expected value of the project effectively and efficiently." The former goal is meaningful only when the quasi-market mode is in operation. Thinking more generally, the objective is to erect and maintain a way to share information with the user so that investments in information technology will be made judiciously. Enough information must be provided for both parties to make good choices on a timely basis and at least cost. A proper match between project characteristics and governance mechanism results in the proper generation of information needed to govern the investment decision. A mismatch, on the other hand, will be costly, generating either insufficient or irrelevant information. Figure 3 summarizes this contingency relationship.

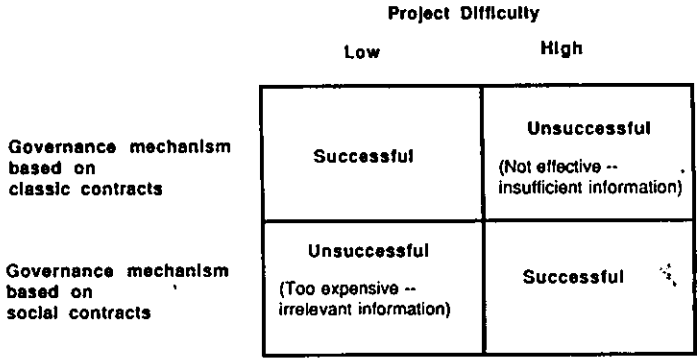

Figure 3. A General Model for Managing the User Relationship

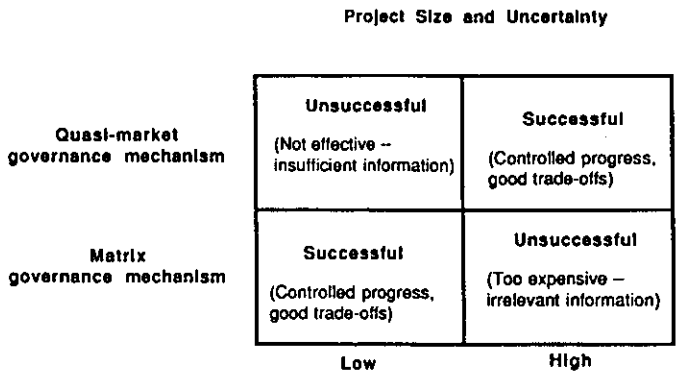

Figure 4. An Exploratory Model for Managing the User Relationship

The general argument is that while low difficulty projects are amenable to governance with classic contracting approaches embedded in the SDLC, more difficult projects require an approach which matches difficulty in specification to a set of acceptable social contracts. That is, the SDLC approach may be a good way to govern some projects. Another way, matrix governance, relies on user involvement to provide a foundation for hammering out social contracts and is better for the tough projects. It seems likely that some IS project managers sense this and behave accordingly. Hence, references are made to matrix-type techniques in the IS literature (Beath 1983), and we might expect to find IS project managers using matrix governance techniques, but without a framework which legitimatizes them or focuses them appropriately. The model proposed here attempts to rectify this situation.

THE STUDY

In the expectation that some IS project managers were already using matrix approaches, a field study was conducted to explore a reduced version of the model in Figure 3 (Beath 1986). The governance mechanisms of interest in this initial exploration were quasi-market and matrix; the project characteristics of interest were project size and uncertainty. Thus, the exploratory model (Figure 4) is more limited than the general model. The field study examined fourteen recent projects at an oil refining and marketing company, called ORM here, using both quantitative and qualitative techniques. The projects were typical MIS applications, emphasizing development of online applications in support of a full range of business functions. Of the fourteen projects, four were selected for in-depth examination and exploration of the model.

This paper uses those four cases to explore three questions: 1) Is proper matching of governance mechanism to project difficulty necessary for effective governance of the project/user relationship? 2) Can difficult projects be effectively managed with matrix governance? 3) What are the consequences of a mismatch? The case study approach is particularly appropriate in this situation, as the phenomena of interest are quite complex, difficult to remove from the work context and occur in small numbers, and the model being explored is still in the inductive stage. The objectives of the research are to enrich the model and to determine whether to pursue testing it using more costly methods.

The information systems organization of ORM is typical of many systems development departments in large corporations. The operating environment is IBM and IBM-compatible; COBOL is the principal application system language. IS reports to the financial vice president, and the systems development group's functional organization mirrors ORM's organization. At the time of the study, development projects were measured on schedule and cost performance, with schedule performance receiving the most attention.

Policies on governance of the project/user relationship at ORM could be characterized as laissez-faire in many respects. Project authorizations were required for development projects requiring more

than six months to complete, but some authorizations were signed just prior to completion. Systems development standards were embodied in SDM/70 (SDM/70 no date), a purchased set of forms and methods espousing the typical SDLC or quasi-market approach to governance of the project/user relationship. But project managers were given the discretion to use SDM/70 as they saw fit; typically they searched the material for methods, forms and techniques they felt would be useful on a particular project. No policy governed user involvement on the projects studied. User liaisons were frequently designated on projects, but the responsibilities of user liaisons varied considerably. All project managers were IS professionals.

Overall, it appears that at ORM, as in many similar IS departments, project management is considered an art, not a discipline. Deviations from strict SDLC norms are tolerated at ORM, especially in the blurring of phases, informal accommodation of changes, and manipulation of budget and schedule records. Thus, it was expected that some projects at ORM might use a project/user governance mechanism that deviated from the usual quasi-market approach, incorporating or substituting matrix mechanisms.

Project/user relationship governance on the fourteen projects in the study was evaluated using interviews with project managers and reviews of project documents. Key indicators of *social contracting* at ORM were reliance on social assessments for progress and completion, the extension of service beyond phase termination date, and delays in signing project authorizations. Particular attention was also paid to *user involvement* as an important supporting mechanism for social contracting. Key discriminators of user involvement at ORM were the degree of responsibility for project activities taken by the user, inclusion of project involvement in user's performance reviews, and the "earliness" of user involvement (late involvement does not provide a foundation for agreements).

Among the fourteen projects examined the typical project lasted about fifteen months, and cost about $600,000. Indications of project relative size and uncertainty were drawn from the project's risk analysis, a form completed at the beginning of each project similar to one described by McFarlan (1981; Dallas Tire Case 1980). Measures of size, technological complexity and social complexity were computed. *Size* included measures of the number of work months, the elapsed time, and the number and size of subsystems to be developed. *Technical complexity* included consideration of mode (batch or online), degree of distribution of process or data, and the number of computers and peripherals involved in the system. *Social complexity* included measures of the severity of procedural changes required in the user area, the number of users, the newness of the technology to the user, the attitude of the user towards computing and the commitment of the user management to the new system.

All fourteen projects were officially on time, reflecting, no doubt, the emphasis on schedule performance; budget variance ranged from +13% to -12%. To measure the adequacy with which the project/user relationship was managed on these projects, ORM's six systems and programming managers were asked to rate each project on six attributes. Among these were two attributes relating to the objectives of the project/user governance mechanisms: (1) display controlled progress -- throughout the project, the status of the project schedule, budget and features to be delivered are communicated clearly to all stakeholders and (2) make good tradeoff choices--between cost, schedule and requirements, particularly with respect to technical issues of design, development and implementation. Recall that the objective of the project/user governance mechanism is to exchange cost and benefit information so that appropriate tradeoff decisions can be made.

Of the fourteen projects initially examined, four were selected for comparative examination of the exploratory model. The projects were selected to represent a variety of relationships between outcomes, governance mechanism and project difficulty. Two "successful" projects (rated second and third by the managers) and two "unsuccessful" projects (rated twelfth and thirteenth) were selected. (Documentation on the best and worst projects was much more limited.) The two successful projects appeared to be examples of appropriate uses of quasi-market and matrix governance, respectively, and they are called Good-Q and Good-M, below. The two unsuccessful projects, called Bad-Q (apparently quasi-market governance) and Bad-? (apparently no governance) provided an opportunity to explore the limits of the model. Figure 5 indicates where the four cases fit into the model.

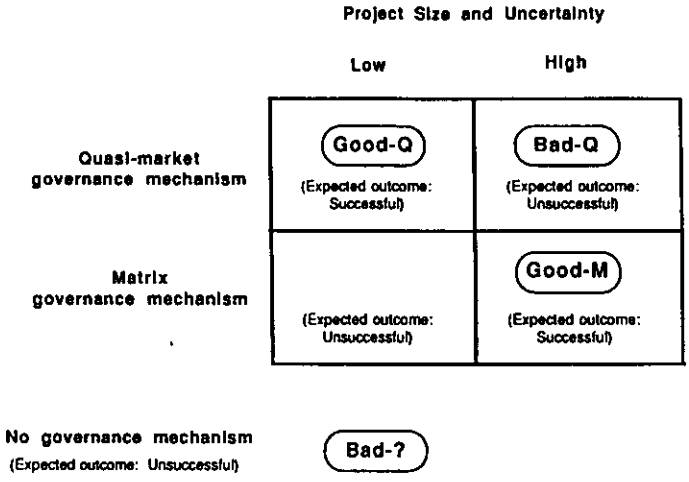

Figure 5. Summary of Case Results

Case descriptions of the four projects were developed from previously obtained material and supplemented by additional interviews with project personnel and other observers and by examination of project documents, such as project documentation and Project Closing Reports (PCR's). The four cases are described below, followed by a discussion of the lessons from the cases for governance of the project/user relationship.

THE CASES

The Second Rated Project -- Good-Q

This highly rated project seemed at first to be a straightforward example of appropriate use of quasi-market governance in a low uncertainty situation. Closer examination, however, reveals some interesting twists.

According to its Project Closing Report (PCR), "The objective of [Good-Q] was to provide Retail Marketing with an on-line computer system for tracking and reporting expenditures against estimated and committed amounts." The new system replaced an existing manual process.

Low Uncertainty. Compared with other projects at this site, Good-Q was small and had low technical and social complexity. The Good-Q project was completed in seven months and cost $165,000. The new system has one passive link to a financial system (merely reading a file), is logically straightforward, uses familiar, established technology, and is not critical to normal business activities. Retail Marketing, the client, was the only "user," and the new system required neither structural changes to the organization nor significant changes to daily operations. "Overall the Risk Analysis did give the total project a low risk rating, which proved to be accurate," says the PCR.

Quasi-market Governance. Nancy, the project manager for Good-Q, is quite familiar with standard system development techniques, and relies on them. "Good-Q followed the SDM/70 project life cycle methodology and guidelines," says the PCR. Following SDM guidelines, user involvement in the project emphasized information sharing, not responsibility for project tasks. Relevant marketing and accounting personnel were interviewed in the requirements phase. Later these same individuals reviewed a specification document describing the proposed system. A user liaison reviewed results of the acceptance test.

Two project authorizations were signed for this project, both in a timely fashion. The first was signed quite near the beginning of the project covering the development of a specification document; the second was signed early in the development activity. Good-Q was completed on schedule, relative to the plan presented in the second project authorization, and two months *ahead* of original estimates. Similarly, project costs were only slightly over the authorized budget (+3%) and 17% *below* original estimates.

Of the fourteen projects examined in this research, this is the only one on which the *authorized* schedule and budget are shorter and smaller than the *original* schedule and budget estimates. On closer examination, however, it is clear that the definition of "complete" for Good-Q meant that the programs were operational, not that they were being used for business purposes. The following passage from the PCR reflects Nancy's argument that the Good-Q project should be regarded as completed:

> The Good-Q system was signed-off by Retail Marketing on November 14, 1984....Implementation paperwork was submitted on November 15, 1984. The start-up database was loaded on November 20, 1984, using available data....The gradual process of moving programs, JCL, data

> sets, libraries, etc., from test to production took place during the last week of November. The installation was completed by November 29, 1984, and the system run in production on November 30, 1984.

However, user training was conducted a few months following this date, and business use of the system did not begin until well into 1985. Apparently the day-by-day description of the wind-down of the Good-Q project demonstrates Nancy's success in convincing MIS and user management that the project team's responsibility was to get the code running on the computer, and that it was the user's responsibility to use it.

During the interview, Nancy contrasted the effectiveness of SDM/70 for project control with a prototyping methodology, called PDM/80 (PDM/80 1984), being evaluated by the department. "PDM/80 doesn't have rigorous controls. SDM lets you finish one step before moving to the next. PDM/80 is fuzzy. SDM much more clearly defines the difference between phases." She continued, "There has to be that instant in time when you release the system to be in production. PDM/80 misses that."

When asked what contributed to the successful control of progress and the making of tradeoff decisions on Good-Q, Nancy cited "rigorous application of SDM/70 techniques" and "keeping in very close touch with the team, so we could always tell where we were." In her opinion, a good relationship between the project and the user depends on satisfactory completion of the formal agreement, and satisfactory completion of the formal agreement rests on close control of the project team.

The exploratory model indicates that quasi-market governance will successfully mitigate small amounts of project difficulty. It appears that in this case the mitigation occurs in an unexpected way. That is, Nancy began with the assumption that quasi-market governance would be used. Then, to the extent necessary, she retro-fitted the project to be manageable under that constraint, including scoping the work to be done so that it ended with delivery of the programs to a computer. The management of the IS department is familiar with Nancy's style and values it, assigning her to manage smaller, less technically complex projects, in full confidence that they will be completed on time and within budget. In this way, governance mechanism and project characteristics are successfully matched.

The Third Rated Project -- Good-M

On this challenging project, heavy user involvement provided a foundation for social contracting. But, as shown below, the project manager also considers some aspects of the quasi-market approach to be important to project success.

The Good-M project modified the company's customer system to support a new multi-million dollar real-time application, called PMS here, which was under development at the same time. The scope of Good-M included fulfillment of several outstanding enhancement requests, but its driving force was the PMS project. The project authorization's return on investment box says, "Not applicable -- PMS prerequisite." The project was funded, along with the PMS project, by a high level steering committee.

A Difficult Project. The Good-M project lasted twenty months and cost just under $1,000,000; at this site, that is a large project. Technical complexity was very high; PMS was the company's first attempt to implement a real-time system. Social complexity was also high, due in part to heavy interdependence with the PMS project. Beyond PMS, two additional groups of users were impacted by the Good-M project: (1) I/O Control, who gave up responsibility for maintaining the customer database, and (2) people throughout the company who took on this responsibility. This group, including people at field offices, regional marketing offices, zone offices, refineries, and headquarters, was represented on Good-M by a set of "coordinators."

Matrix Governance. The Good-M project had very high user involvement on several fronts. Good-M project team members met daily with representatives of the main PMS development team and two other PMS support projects. In effect, Good-M had "users" who were capable of and interested in participating in detail design and programming. Standards and data definitions were shared, and the bulk of the planning for testing and installation were carried out jointly. A user liaison (from I/O Control) and the coordinators were heavily involved in the requirements definition, systems test and implementation of the new customer database. The

coordinators were given the opportunity to try to "crash" the system for about three weeks, which provided a healthy measure of exercise for the new system. The user liaison led the training process.

The Good-M project apparently made extensive use of social contracting within a framework of a single large project authorization. No project authorization covered the development of requirements, a violation of local standards. The single project authorization, for $1.1 million, was signed after the requirements were completed. Subsequently, however, changes to requirements needed for PMS were accommodated throughout the project. The PCR says, "In September, 1983, the requirements were frozen." The PCR then goes on to describe a lengthy series of changes originating from the PMS project. Finally, it says, "During the final integration and acceptance testing phase, changes were still being received from the PMS teams."

Good-M's project authorization does not include a schedule; of the project authorizations reviewed, this is the only one without a proposed delivery date. The PCR, however, makes reference to an "original" plan and a "revised" plan. The PCR says,

> The Programming and Implementation Phase was originally scheduled to be completed in April, 1984. However, due to the many uncertainties surrounding the new PMS environment and the continuous changes requested by both the PMS teams and on-going business operations, the implementation phase was extended to July 12, 1984.

The project manager said that the revised schedule was agreed to by "all parties" just prior to the new July due date, well after the original April due date. This agreement notwithstanding, department performance reports record Good-M complete as of December, 1984, "on time."

Pinpointing the cost performance of Good-M is equally difficult. The total costs of the project are reported to be 10% *under* budget. There is also a note that about 8.6% of the total charges for Good-M are for "PMS support." The PCR also notes that much overtime -- roughly equivalent to the 10% underrun -- was needed to meet the schedule.

Oscar, the project manager, discussed governance of project/user relationships. In his view, user involvement can be difficult to obtain for several reasons. First, as in this case, users may be geographically dispersed. Second, the most knowledgeable users have their own jobs to perform. Third, users may fear automation, or may have had bad experiences working with analysts in the past. So Oscar says that one must work very hard to establish *trust* with the users. On this project, he visited numerous user sites, watched and listened during the day, and took people to dinner, listening some more. "You have to know who they are and how they communicate," he says. He spoke frequently on the phone with users, "but nothing beats one on one."

Oscar is motivated by a generalized notion of investing in future payoffs when he establishes relationships with users. When other analysts are only gathering requirements information, Oscar is "building trust." On this foundation, informal agreements can be made easily and quickly according to Oscar. Oscar also stressed the importance of formal agreements. "Each has its own place," he says. "They make up a system of checks and balances. Things like requirements documents keep people honest. Without that you'd get so sloppy, it would be chaos." Overall, in his view, formal agreements set the limits for the investment, user involvement provides an opportunity for mutual trust, and the trust allows the details of the investment to be worked out informally.

The Good-M project and Oscar's philosophy support the model. Oscar achieves success by using SDLC-type authorizations and documents for the organizational legitimacy they provide and as an umbrella under which he works more informally, relying on personal relationships, mutual trust, and shared understanding of goals.

The Twelfth Rated Project -- Bad-Q

This project was relatively difficult, and the inexperienced project manager hoped that rigorous application of the quasi-market governance techniques she had learned would minimize the consequences of that difficulty. Unfortunately, things did not work out. Maintenance costs on this system were high, and the system suffered from complexity due to late design changes. The project also apparently suffered from bad public relations.

The objective of the Bad-Q project was to replace a batch operational control system used at the refineries with an enhanced online/database system. The project was funded by the headquarters' Refining Department, as are all data processing systems used at the refineries. For several years headquarters had been attempting to make the administrative processes used at the refineries more uniform. Common computer systems are seen as one way of achieving this objective. In addition, the business users affected by this system included not only the refineries, but also marketing field offices.

Relatively Difficult. The Bad-Q project started in June, 1981, and took twenty months to complete at a cost of about $635,000. Relative to other ORM projects, Bad-Q is just above average in size. This was the first of a series of similar replacements, and thus was a proving ground for several new technologies. The project team was relatively inexperienced, leading to high technical complexity. Social complexity of Bad-Q was also high; not only was the user community large and dispersed, but implementation of Bad-Q implied changes to organization structure and jobs at the refineries.

Quasi-Market Governance. In the requirements phase, users at all refineries were interviewed, following SDM guidelines. When the requirements were complete, design reviews were held at the refineries. User involvement was more intense during implementation. Refinery personnel used the new communications network to participate in the system test long distance, so to speak. According to Jane, the project manager, "The network allowed us to demonstrate our system to users in the refineries ....In doing so we were able to give the users first-hand knowledge of the system operation and obtain feedback from them during our testing phase." This involvement was found to be extremely useful in finding errors, but provides little foundation for social contracts.

In the contracting arena, it seems fair to say that Jane, managing her first project, made a sincere effort to conform to quasi-market expectations for project/user governance. Two project authorizations and one development change request were signed for Bad-Q. Bad-Q's was the only development change request signed among the fourteen projects examined in this research. It covered a design improvement suggested by the project team; in the opinion of another manager, only a novice would have handled a design improvement in this manner.

As a result of a scope change at the end of the requirements phase (from batch updating to online updating), the requirements had to be partly redone, and the first project authorization was officially overrun. (A more experienced manager might have included the additional work in the next phase.) Increases to cost and schedule of subsequent phases required by the increased scope were included on the second project authorization. In the end, however, the project was nearly 10% over authorized budget, with the overrun about equally divided between the two halves of the project. Only two of the fourteen projects officially overran their authorized budgets by more (by about 12% in both cases). Little effort was made to hide the overrun. "I know a lot more now," Jane said, ruefully.

When asked what went wrong, Jane replied, "This project didn't get the attention it deserved, from the refinery, from marketing, from IS." She contrasted this with her next project, which was over twice the size: "On MMM we had so much user involvement. It was wonderful. You need commitment and energy from the users."

At the end of the project, in Bad-Q's PCR, Jane made a number of recommendations for future refinery-based implementations. Some involved the use of tools and techniques, but many others can be interpreted as a call for larger investments in establishing matrix governance mechanisms. With respect to user involvement she said, "If this type of cross-functional effort is to be undertaken in other projects, it is my recommendation that specifications be developed with heavy user involvement from all organizations." And elsewhere, "it is important to train both users and data processors in order to facilitate [future] refinery implementations." Generally, she advocates a longer, fuzzier implementation schedule, with some burn-in time at the first site, or a "post-implementation follow-up" -- a sort of mop-up operation. These make the delivery date harder to establish and violate quasi-market principles. Finally, she advocates more involvement by the refinery computing staff throughout development (as opposed to just during the installation of the programs) "in order to cut travel costs on future projects." That is, raising the level of expertise of refinery computer operators on one project could reduce development costs on subsequent projects.

In terms of the model being examined, it seems likely that the problems on Bad-Q stemmed from insufficient early user involvement and too much reliance on quasi-market governance to deal with uncertainty. The model suggests that attempting to govern a difficult project with quasi-market mechanisms will result in a less than optimal return on investment, owing to suboptimal tradeoff decisions and increased costs of rework. Evidence of both are indicated in this case.

The Thirteenth Rated Project -- Bad-?

Apparently neither matrix nor quasi-market approaches were used on this small and relatively straightforward project. The outcome, not unexpectedly, was unsatisfactory.

The Bad-? project developed an online system to maintain and access data common to several crude oil supply systems. The new system consists of several data tables (containing the common data) and a rule database (governing the maintenance of and access to the data). The project was initiated and funded by a steering committee of managers in the crude oil supply department. Two other projects under development at the same time were expected to use the Bad-? database.

Relatively Easy. Bad-? was completed in eleven months and cost about $230,000. The project team included the project manager, an independent contractor with excellent technical skills, known as "the Guru," and two other programmer/analysts. The project was small, and its technical complexity was low (particularly in the hands of the Guru), as was its social complexity. The principal users of the system were the members of the two teams expecting to use the Bad-? database. Other users were those who would maintain the file entries and rules, but as the files were small and stable, there was little impact on the task environment of these users.

No Project/User Governance Mechanism. User involvement on this project was very low. The design of the interfaces between Bad-? and the systems expecting to use its database was left mostly to the Guru, as it was felt that most of the constraints were at the Bad-? end of the interface. The business users attended a presentation on the design, but did not review the specification document itself. "During programming," said Sara, the project manager, "the Guru was closeted."

Relations between Sara and the Guru were difficult. Sara is a business analyst, not a technician, and the Guru is a "brilliant, private, arrogant genius," according to one observer. They did not communicate. In talking about how change control was handled, Sara commented "After we agreed on a change, he just changed the code, or made a note in the online file of system documentation. I never saw anything." From her point of view, he was secretive. In one outsider's view, she was not technically strong enough to grasp his designs. Sara admitted freely that she had no idea how the Guru's programs worked.

Sara did not make extensive use of formal contracting on Bad-?. Five months after starting work and six weeks before the projected implementation, a project authorization for $132,000 was signed. It appears that, at that time, this was expected to be the only project authorization for the entire project. However, a few weeks later, when most, but not all, of the programming was complete, the Guru became involved in a dispute with management over another issue and quit. Two months later (after the expected implementation date), a supplementary project authorization for $90,000 was obtained. It stated "A significant portion of this additional work has been required due to the departure of a key contract analyst at a critical point in this project" [underscore in original]. The remainder of the team finished the project within the new schedule and budget.

Because of the delivery delays on Bad-?, one of the two projects which were supposed to use the common database had to build its own files. The second project was scrapped for other reasons. Some subsequently developed systems use Bad-?, but it was the opinion of one manager that Bad-? served little useful purpose. Very little of the data was common, and the effort involved in adding new tables to the Bad-? system is roughly equivalent to putting a table into an application. Maintenance costs are high.

Sara thinks that the problem on Bad-? was that the Guru tried to build an overly complicated system, "to try to provide too far out into the future." The Guru's objectives for the system and the company's objectives for the system, in her opinion, were not in sync.

Governance of the project/user relationship on Bad-? seems to be characteristic of neither matrix nor quasi-market approaches. Rather, user apathy combined with a suspension of normal project controls boils down to no governance. Furthermore, it appears that governance of the user relationship was undermined by an absence of governance within the project team. That is, without sufficient familiarity with the costs and features of the system, Sara could not communicate those to her user community, and could not suggest appropriate tradeoffs.

A quasi-market approach might have been appropriate for this project, according to the model being tested here. In fact, this seems to have been a situation in which a formal contract with the Guru, or pure market governance, would have been desirable.

SUMMARY

Examination of the Good-Q and Good-M projects confirms that these two cases conform to the model's predictions. Concerning Bad-Q, over reliance on quasi-market approaches penalized both Bad-Q and future refinery projects because design errors were made and opportunities for investments in matrix governance were forgone. Bad-? appears to have suffered from an absence of governance of the project/user relationship. Without governance, cost overruns and specification failures are not unexpected.

While the use of a convenience sample precludes drawing any conclusions regarding the accuracy of the model, these cases, taken as a group, offer preliminary indications that the model warrants closer examination. It does appear that governance mechanism and project difficulty should be matched for effective governance. The appropriateness of using the two governance mechanisms being evaluated -- quasi-market and matrix -- for dealing with low and high project difficulty, respectively, is supported. The consequence of a mismatch between quasi-market governance and project difficulty is a predictable lack of effectiveness.

Most of the MIS literature emphasizes the design improvement potential of user involvement and the SDLC. But, here we see that both of these are also tools of governance, that each one has a limited range of effectiveness, and that neither is universally useful. As the uncertainty of projects undertaken in MIS departments increases (with the easy projects undertaken by users and information centers or with purchased software), more effective mechanisms for managing difficult projects will be required. The results of this research suggest that part of the solution is judicious user involvement and mastery of social contracting techniques.

The cases also show how difficult it can be to make an intra-organizational development situation conform to quasi-market expectations. In only one case (Good-Q) were costs (in the project authorization) and features (in requirements documents) controlled in the same organization. In two cases (Good-M and Bad-?), primary users were within IS, while contracts were with steering committees. In Good-M and Bad-Q, users were hierarchically, geographically and functionally dispersed. In no case were the future maintainers of the systems considered "users." Wearing quasi-market blinders, we see only those users with whom we have formal contracts, and simply ignore all others. There is no way to take them into account.

The cases show that IS project managers need to master a variety of governance techniques, which can be matched to a variety of exchange characteristics. The project can be seen as the nexus of many exchanges, each requiring some kind of governance, with users who finance projects, users who submit or maintain data, users who access the data, and also with data processing operators, data control clerks, program or database maintainers, and IS management. As IS is called upon to build more information systems that provide strategic linkage across subunits, the "virtual user" approach will become increasingly unsatisfactory.

The mastery of social governance techniques requires, first, their legitimation. This research motivates that legitimation process. The project managers interviewed for this study all agreed that social contracting was both useful and necessary, but none seemed to think it was quite "right." Some IS managers dismiss efforts to build trust with users as wasteful "schmoozing." By studying the appropriate role of social contracting, we can learn to apply these techniques in a controlled, organizationally acceptable manner.

One important step in the legitimation process is to clarify objectives for the project/user relationship. This study emphasizes the view that managing that

relationship means locating and sharing the cost and feature information needed to manage an investment in computing. Delivery of a particular product, at some prespecified cost and schedule, is a reasonable objective only if the relationship is amenable to quasi-market governance.

Another important step in the legitimation process is to distinguish between the costs to *establish* a governance mechanism (roughly speaking, a one-time cost), and the costs to *operate* a governance mechanism (e.g., recurring costs). At present, it is difficult to separate those costs on IS projects. Our quasi-market habits make us blind to opportunities for investing in future governance. Similarly, we fail to encourage users to make similar investments, on their side. Making optimal governance choices rests on having much more information about what the relative costs to establish and operate governance mechanisms are under different circumstances, and having a clearer picture of the opportunities for recontracting.

Finally, the cases suggest that the model oversimplifies the typical within-organization IS project management situation. The Good-M case reminds us that the organizational distance between IS and its clients, reinforced by specialization and professionalization within IS, makes it very difficult to get completely away from using at least some quasi-market approaches. Good-M also shows the important relationship between user involvement and social contracting. And, as shown in both Bad-Q and Bad-?, without thoughtful user involvement, meaningful social contracts are very hard to conclude satisfactorily. Good-Q and Bad-? expose the futility of trying to study the effectiveness of the project/user governance mechanism without including consideration of the way in which the team was governed. These four cases suggest that project manager experience (or sensitivity to local norms favoring quasi-market governance) may be more influential in the selection of governance mechanism than project difficulty.

While the model does seem promising, much additional research is necessary to fully articulate the matrix governance approach in IS projects. These few cases demonstrate only that this additional work is warranted. The elaboration of the model which surfaces in the cases is an important side-benefit to case studies, particularly since the research is still in its inductive phase. More narrowly defined exchanges should be examined from both the user and IS perspective. The impact of past and future exchanges on project difficulty should be specifically addressed in future research. Finally, the range of effectiveness of other governance options, such as classical contracting, should be pursued.

REFERENCES

Barney, J. B., and Ouchi, W. G. (Eds.). *Organizational Economics.* Jossey-Bass, San Francisco, 1986.

Beath, C. M. "Strategies for Managing MIS Projects: A Transaction Cost Approach." *Proceedings of the Fourth International Conference on Information Systems,* Houston, TX, 1983, pp. 133-147.

Beath, C. M. "Managing the User Relationship in Management Information Systems Projects: A Transaction Governance Approach." Unpublished doctoral dissertation, University of California, Los Angeles, 1986.

Boehm, B. W. *Software Engineering Economics.* Prentice-Hall, Englewood Cliffs, NJ, 1981.

Dallas Tire Case, Harvard Business School Case Number 9-180-006, HBS Case Services, Harvard Business School, Boston MA 02163.

Davis, G. B. *Management Information Systems: Conceptual Foundations, Structure and Development.* McGraw-Hill Book Company, New York, 1974.

Ives, B., and Olson, M. H. "User Involvement and MIS Success: A Review of Research." *Management Science,* Vol. 30, No. 5, May 1984, pp. 586-603.

Markus, M. L. "Power, Politics and MIS Implementation." *Communications of the ACM,* Vol. 26, No. 6, 1983, pp. 430-444.

McFarlan, F. W. "Portfolio Approach to Information Systems." *Harvard Business Review,* Vol. 59, No. 4, July-August 1981, pp. 142-150.

Ouchi, W. G. "Markets, Bureaucracies and Clans." *Administrative Science Quarterly,* Vol. 25, No. 1, March 1980, pp. 129-141.

PDM/80, D. Appleton and Co., Inc., 1334 Park View Ave., Suite 220, Manhattan Beach, CA 90266, 1984.

Powers, R. F., and Dickson, G. W. "MIS Project Management: Myths, Opinions and Reality." *California Management Review*, Vol. 15, No. 3, 1973, pp. 147-156.

SDM/70, Atlantic Software, Inc., Lafayette Building, Fifth and Chestnut Streets, Philadelphia, PA 19106 (no date).

Swanson, E. B. "Management Information Systems: Appreciation and Involvement." *Management Science*, Vol. 21, 1974, pp. 178-188.

Williamson, O. E. *Markets and Hierarchies: Analysis and Antitrust Implications.* The Free Press, New York, 1975.

Williamson, O. E., and Ouchi, W. G. "The Markets and Hierarchies and Visible Hand Perspectives." In A. H. Van de Ven and W. F. Joyce (eds.), *Perspectives on Organization Design and Behavior*, John Wiley and Sons, New York, 1981.

Zmud, R. W. "Management of Large Software Development Efforts." *MIS Quarterly*, June 1980, pp.45-55.