2008

Project Quality of Offshore Virtual Teams Engaged in Software Requirements Analysis: An Exploratory Comparative Study

Dhruv Nath
Management Development Institute

Varadharajan Sridhar
Management Development Institute

Monica Adya
Marquette University, monica.adya@marquette.edu

Amit Malik
Management Development Institute

IGI PUBLISHING
701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.igi-global.com

ITJ4388

This paper appears in the publication, Journal of Glabal Information Management, Volume 16, Issue 4
edited by **Felix B. Tan © 2008, IGI Global**

Project Quality of Off-Shore Virtual Teams Engaged in Software Requirements Analysis:
An Exploratory Comparative Study

Dhruv Nath, Management Development Institute, India

Varadharajan Sridhar, Management Development Institute, India

Monica Adya, Marquette University, USA

Amit Malik, Management Development Institute, India

ABSTRACT

The off-shore software development companies in countries such as India use a global delivery model in which initial requirement analysis phase of software projects get executed at client locations to leverage frequent and deep interaction between user and developer teams. Subsequent phases such as design, coding and testing are completed at off-shore locations. Emerging trends indicate an increasing interest in off-shoring even requirements analysis phase using computer mediated communication. We conducted an exploratory research study involving students from Management Development Institute (MDI), India and Marquette University (MU), U.S.A. to determine quality of such off-shored requirements analysis projects. Our findings suggest that project quality of teams engaged in pure off-shore mode is comparable to that of teams engaged in collocated mode. However, the effect of controls such as user project monitoring on the quality of off-shored projects needs to be studied further.

Keywords: *collocated teams; computer-mediated communications; off-shoring, project management; project control*

INTRODUCTION

The past two decades have witnessed significant globalization of the software development process with development rapidly moving away from the traditional collocated model, often called *on-site development*, to the off-shoring model. With the availability of increasingly skilled, flexible, and economical IT workforce in countries such as India, Malaysia, and China, it makes financial sense for United States and

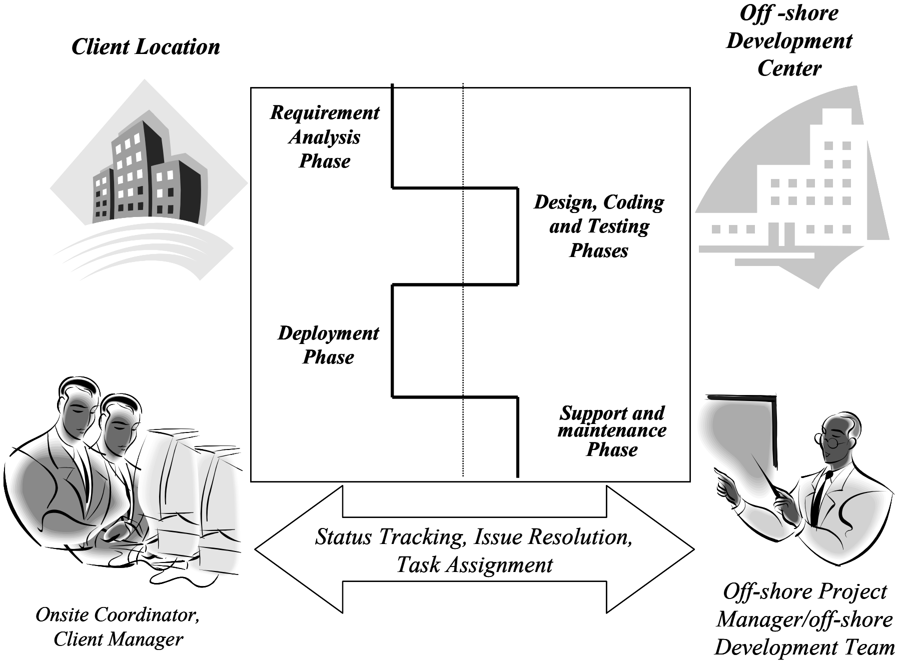
European client organizations to execute a significant portion of software projects in these countries. This growing trend towards off-shoring has, in turn, spurred growth in many Asian nations, creating improved economic and IT infrastructure and enhancing the viability of these countries as software service providers. For example, India has emerged as a dominant off-shore software development industry with revenue of about $16.7 billion, which is projected to reach $60 billion by the year 2010 (Carmel, 2006; National Association of Software and Service Companies, 2005).

The Indian off-shore software industry has matured over the years, and process capability has been steadily improving. Coordination and communication problems typically encountered in off-shore development (see Battin, Crocker, Kreidler, & Subramanian, 2001, for an extended discussion), are mitigated by the use of processes such as rational task assignments and liaisoning, and tools such as centralized bug reporting system and software configuration management platforms. A case in point is India's Infosys Technologies, which has significantly leveraged time zone differences with its clients by modifying its organizational culture, processes, and communication technologies (Carmel, 2006).

The typical off-shore development model, followed successfully for over a decade by many Indian software companies such as Infosys, Wipro, TCS, and Satyam, is illustrated in Figure 1.

Requirements analysis refers to that stage of the system development life cycle wherein the information and information processing services needed to support select objectives and functions of the organization are (i) determined and (ii) coherently represented using well-defined artifacts such as entity-relationship diagrams, data-flow diagrams, use cases, and screen prototypes (Hoffer, George, & Valacich, 1999). As suggested in Figure 1, typically this

Figure 1. The off-shore software development model



phase is conducted at the client location, since this phase requires frequent and significant interaction between users and developers. Business and systems analysts are physically located at the client site to perform this activity. Global projects consultant teams from off-shore location travel to the user site to gather and analyze requirements in face-to-face meetings (Damian & Zowghi, 2002). The consultants then communicate the requirements to the development staff in the offshore site. Depending on the nature of the project, high-level design is conducted in both on-site and off-shore mode due to comparatively lower interaction needs with the client. Detailed design, coding, and testing are executed at the off-shore site. Off-shore vendors also deploy liaisons who coordinate activities between on-site users and the off-shore development team. These liaisons are critical for effective communication and coordination between users and developers (Battin et al., 2001).

Increasingly, both client and software providers are now considering the possibility of off-shoring the requirements analysis phase, traditionally done on client site, away from the client location. In such a scenario, analysts and developers located at the off-shore location would interact in a virtual mode with the clients situated at their premises to determine and structure the requirements. Such a shift could potentially improve the cost arbitrage of the projects for instance by cutting down travel costs incurred for sending analysts to the client site for face-to-face meetings. In an extreme case, the entire team of analysts and developers could be based in off-shore location such as India while the client could be in Europe or the United States. Requirements gathering would then be conducted between these virtual teams using existing computer-mediated communications such as chat, e-mail, and video conferencing. The questions of research interest then are:

1. Can requirements analysis conducted by collocated teams using face-to-face communication be comparable or better than those produced by virtual off-shore teams using computer-mediated communication?
2. What forms of control are necessary to facilitate high-quality outcomes from virtual requirements analysis undertakings?

Using theories of social presence, media richness (Burke & Chidambaram, 1999), as well as control theory (Kirsch, 2002), we develop and test hypotheses regarding these questions.

Traditionally, user involvement in IS projects has been an important contributor to project success (Hartwick & Barki, 1994; Foster & Franz ,1999; Lin & Shao, 2000; Sridhar, Nath, & Malik, in press). Lack of user proximity in a virtual setting can potentially limit the quality of requirements elicitation due to limitations of communications media. In order to mitigate these limitations and the absence of analysts and developers at customer premises, user involvement is expected to take the form of close project monitoring and control to ensure that requirements and project goals are met. Control theory provides the required theoretical foundations for analyzing the effect of different types of controls on teams (Crisp, 2003). In this study, we specifically consider user project monitoring as a behavioral control mechanism and examine its impact on project quality during requirements analysis phase of off-shored software projects. Further, we explore the intersection of media richness and control theories to find early answers to the research questions raised earlier.

This study is exploratory in nature. Without loss of generality, we restrict our attention to the requirements analysis phase as defined in the structured systems analysis and design (SSAD) methodology as defined by Hoffer et al. (1999). We define requirements analysis as subsuming the following two phases:

1. Requirements determination: The process by which the analysts determine the requirements of the system from the users through discussions and interviews and exchanging forms, reports, job descriptions and other necessary documents.

2. Requirements structuring: The process by which the analysts coherently represent the information gathered as part of requirements determination using process modeling and logic modeling tools as described in SSAD.

Our interactions with managers in client firms engaged in software development indicate that off-shoring of requirements analysis is still uncommon. Hence it is not practical to analyze this phenomenon of pure off-shoring of requirements in real-life setting. It is also difficult to do in-depth longitudinal or cross-sectional case studies. Given these arguments, an exploratory research study was conducted in an academic setting involving management students enrolled in a graduate-level information systems course at Management Development Institute (MDI), India, and management students enrolled in a graduate level IT Project Management course at Marquette University (MU), United States. MU students role-played as virtual users/project managers while MDI students were software developers for MU teams as well as user clients for collocated MDI teams. Prior to a full description of our undertaking, we first discuss existing literature on virtual teams in software projects. We then describe the theoretical foundations of this study and elaborate on our research design. Next, we discuss our measures and discuss study outcomes. The article concludes with implications for future research in this context.

VIRTUAL TEAMS IN SOFTWARE PROJECTS

In a pure off-shore mode, users at the client location and the developers at the off-shore location never meet face to face and hence operate as virtual teams, primarily linked through technology across national boundaries. It is in this context that we review previous research on such virtual teams, specifically those engaged in software development projects. Virtual teams are becoming the norm in most corporate environments such as consulting firms, technology products, and e-commerce (Lurey & Raisinghani, 2001) and are being increasingly examined in academic literature (see Powell, Piccoli, & Ives, 2005 for a comprehensive survey of virtual teams). Battin et al. (2001) described how Motorola deployed global virtual teams across six different countries for a Third Generation Cellular System product development. Software development in Alcatel was handled by a central group of several thousand engineers distributed throughout the world (Ebert & De Neve, 2001).

Few studies however, have, examined the use of virtual teams for requirements analysis. Edwards and Sridhar (2005) studied the effectiveness of virtual teams in a collaborative requirements analysis practice. In that study virtual teams at near and far locations participated in requirements analysis phase of the project. This typically is applicable in collaborative global product development exercises as described in Battin et al. (2003). In contrast, in this study we look at the requirements analysis phase of off-shored software projects in which the two protagonists are (i) users who specify the requirements, and (ii) developers who determine and document these requirements together constituting a collaborative virtual teams. Damian and Zowghi (2002) studied the interplay between culture and conflict and the impact of distance on the ability to reconcile different viewpoints with respect to "requirements negotiation" processes. They found that lack of a common understanding of requirements, together with reduced awareness of local context, trust level, and ability to share work artifacts significantly challenge effective collaboration among remote stakeholders in negotiating a set of requirements that satisfies geographically dispersed customers. Damian, Eberlein, Shaw, and Gaines (2000) examined the effect of the distribution of various stakeholders in the requirements engineering process. They found that highest group performance occurred when customers were separated from each other and collocated with the facilitator or system analyst. Our study further contributes to the literature on virtual teams engaged in off-shored software requirements analysis.

THEORETICAL FOUNDATIONS AND HYPOTHESES DEVELOPMENT

Social Presence and Media Richness Theories

Social presence is the extent to which one feels the presence of a person with whom one is interacting. Short, Williams, and Christie (1976) suggested that some media convey greater social presence than others. For instance, face-to-face interaction is considered to be high in social presence, primarily because of the capacity of the medium to transmit proximal, facial, and other nonverbal cues relative to other media. In contrast, computer-mediated communication such as e-mail exhibit inherently lower bandwidth than face-to-face interaction, thus permitting transmission of fewer visual and nonverbal cues and restricting socio-emotional communication (Rice & Love, 1987). In addition to differences in social presence, media richness theory proposes that, given their limited cue-carrying capacity, leaner media such as e-mail, will be less effective for groups performing ambiguous tasks which require a variety of cues to be exchanged. However, Burke and Chidambaram (1999) pointed out that despite some support for media characteristics-dependent theories, overall empirical evidence has been mixed.

Quality of Off-Shored Projects vs. Collocated Projects

Teams engaged in pure off-shored projects primarily rely on computer-mediated communications (synchronous such as chat, audio and video conferencing as well as asynchronous such as e-mail) for interaction. However, collocated teams have the luxury of rich face-to-face communication. Based on the social presence and media richness theories, we formulate the following hypothesis:

H1: *Collocated teams using face-to-face communication will produce higher quality project artifacts compared to virtual teams using computer-mediated communication during the requirements analysis phase of software projects.*

In a subsequent section, we define *quality of project artifacts* and how it is measured. To the best of our knowledge, quality of projects and performance of virtual teams engaged in the software requirements analysis has not been studied in the literature thus far. Although several researchers have compared performances of traditional collocated teams with that of virtual teams, the conclusions have been mixed. While one study reported greater effectiveness for virtual teams (Sharda, Barr, & McDonnell 1988), others such as McDonough, Kahn, and Barczak (2001) have found that virtual teams could not outperform traditional teams. Andres (2002) reported that teams working in face-to-face settings experienced greater productivity compared to those supported using videoconferencing. Generally, computer-mediated teams exhibit lower frequency of communication than face-to-face teams, although they tend to exchange more task-oriented messages as a proportion of total communication (Burke & Chidambaram, 1999; Chidambaram, 1996). This enhanced communication leads to comparable or even higher performance of virtual teams as compared to collocated teams (Burke & Chidambaram, 1999). Consistent with these findings, Schmidt et al. (2001) reported that virtual teams are more effective in new product development decisions as compared to face-to-face teams. However, a majority of the early work has detected no difference between the two types of teams (Burke & Aytes, 1998). Other studies have found no significant differences between traditional and virtual teams when examining decision quality (Archer, 1990; Chidambaram & Bostrom, 1993) as well as the number of ideas generated by decision making teams (Archer, 1990; Lind, 1999; Sharda et al., 1988). Walther (2005) further suggested that complex human processes such as negotiation actually improve between physically distributed individuals who communicate using media low in richness. Studies comparing performance of virtual and collocated teams in software requirements analysis phase are even fewer. Damian et al. (2000) found that groups in face-to-face meetings performed no better than the electronically mediated groups in the requirements negotiation phase of the software development life cycle.

Control Theory

Control is defined as the set of mechanisms designed to motivate individuals to work in such a way that desired objectives are achieved (Kirsch, 1996). *Formal controls* rely on mechanisms that influence the controllee's behavior through performance evaluation and rewards (Choudhury & Sabherwal, 2003). Controllers utilize two modes of formal control: behavior and outcome (Kirsch, 2002). In behavior control, appropriate steps and procedures for task performance are defined by controllers, and then controllees' performance is evaluated according to their adherence to the prescribed procedures. In outcome control, controllers define appropriate targets and allow controllees to decide how to meet those output targets. Controllees' performance is evaluated on the extent to which targets were met, and not on the processes used to achieve the targets (Kirsch 2002).

Informal control mechanisms utilize social or people strategies to reduce goal differences between controller and controllee. Self-control, one mode of informal control, occurs when an individual sets up his or her own goals, self-monitors goal achievement, and rewards or sanctions him- or herself accordingly (Kirsch, 2002). *Clan control,* the other type of informal control, is implemented through mechanisms that minimize the differences between controller's and controllee's preferences by "promulgating common values, beliefs and philosophies within a clan, which is defined as a group of individuals who are dependent on one another and who share a set of common goals" (Choudhury & Sabherwal, 2003). Kirch et al (2002) extended the control theory to the role of client liaisons, exercising control of IS project leaders to ensure that IS projects meet their goals. The study examined the conditions under which client liaisons of IS development projects choose various modes of control. In a related work, Choudhury and Sabherwal (2003) examined the evolution of portfolio of controls over the duration of outsourced IS development projects. They conclude that in outsourced software projects outcome controls are exercised at the start of the project. Behavioral controls are added later in the project. Clan controls are used when the client and vendor had shared goals, and when frequent interactions lead to shared values. Both these studies analyzed the evolution and choice of controls in IS projects and not on the effect of these controls on project outcome. In this study we focus on the effect of formal modes of control (both outcome and behavior) on the quality of project artifacts produced by virtual teams engaged in software requirements analysis. Project monitoring provides opportunities for both forms of formal control previously described through tracking, interpretation and transmission of status information (Crisp, 2003). In this study, we define *user control* to include not only monitoring the project plan (a form of behavioral control) but also the evaluation of the formal artifacts produced (a form of outcome control) during the requirements analysis process. Monitoring of costs is excluded as requirements analysis is often part of a large IS outsourcing project. Though cost monitoring is vital, it does not assume much significance when considered for only one phase of the project and hence is excluded. Based on the control theory and literature review of virtual teams, our second hypothesis is as follows:

H2: *Developer teams that are closely monitored by their users in a virtual team mode will produce higher quality of artifacts as compared to developer teams that are not closely monitored by their users.*

RESEARCH DESIGN

To test both the aforementioned hypotheses, we conducted two overlapping quasi experiments involving students at MU and MDI in controlled settings. Such experimental settings have been actively used in distributed software engineering laboratories and business schools to conduct virtual team exercises in their courses (Powell et al., 2005). A controlled experimental approach provides three benefits. Firstly, it makes available several teams that work in parallel, thereby generating rich data for drawing conclusions. Secondly, it permits researchers to experiment with newer approaches, which may not yet have been explored by the industry. Finally, it equips and trains software engineering students to understand and to handle the challenges of working in global software teams (Favela & Pena-Mora, 2001). A survey on virtual team research by Powell et al. (2005) cited 28 academic experiments and only 13 case study research papers. Our experimental setup is illustrated in Figure 2 and described in greater detail next.

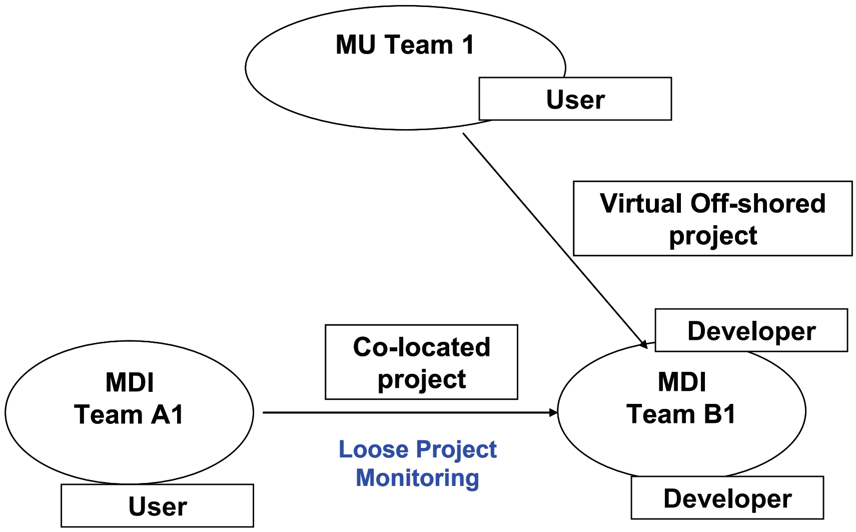
Experiment 1—Testing H1: The Impact of Media Richness on Project Quality

For hypothesis 1 (H1), we compared the quality of projects produced by collocated teams with those that were produced by virtual teams. The collocated teams were students of the post-graduate program in management (equivalent to an MBA) who were attending a core course in management information systems (MIS) at MDI. One hundred and twenty-seven students were divided into two roughly equal sections, section *A* and section *B*. Students from section *A* were grouped into 10 teams of 5 or 6 students each. Each team played the role of users for the collocated project. Figure 2(a) shows one such team, referred to as MDI team A1. Students from section *B* were also grouped into 10 teams of 5 or 6 students each. Each of these teams formed developer teams for the collocated project. Figure 2(a) shows one such team, referred to as MDI team B1. Each MDI *A* team was then paired with one of the MDI *B* teams, as shown in Figure 2(a). Thus MDI team A1 served as users to MDI team B1, the developers in the collocated project. Similarly, MDI team A2 was the user for MDI team B2, and so on.

Setting for the Virtual Teams

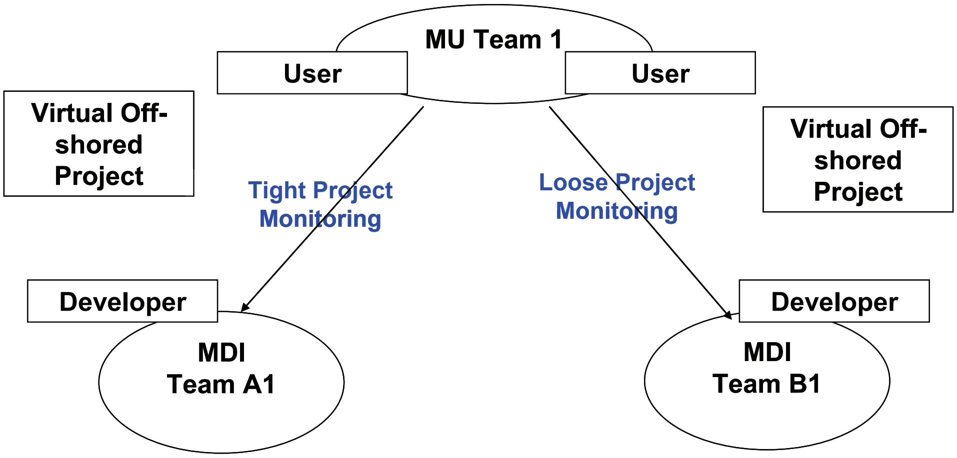
MU students, enrolled in a graduate elective course in IT project management, assumed the role of virtual users. Twenty-eight students divided into 10 teams (each with a team size of 2-3 members), referred to as *MU Teams*.

Figure 2. The experimental set-up



(a) Experiment 1: Collocated vs. Virtual Teams

Figure 2. continued



(b) Experiment 2: Virtual Teams Under Tight Project Monitoring vs. Loose Project Monitoring

Figure 2(a) shows one of these MU teams, team 1. Each MU team was paired with one of the MDI B teams. Thus MDI B teams became the off-shore development teams for the associated MU user teams. These teams consisting of users and developers worked in virtual team mode. In summary, each MDI team *B* was involved in the following two projects: (i) collocated project with MDI user team *A* and (ii) virtual off-shored project with MU user team.

In both projects, the MDI *B* teams were required to submit a project plan at the beginning of the project, detailing various activities and timelines. The final delivery date was predetermined by the instructors based on the course schedule. Project monitoring was voluntary between MDI B and MDI A user teams, so as to minimize the impact of any other variables on the experiment. The MDI B development teams communicated with their corresponding user teams at MU through online means such as e-mail, instant messaging, and voice chats such as Skype and with their MDI A user teams through face-to-face meetings while having face-to-face interactions with their collocated MDI *A* teams. It must be noted that each developer teams (i.e., MDI *B* teams) had 5 or 6 members, thus controlling for the effects of team sizes on the quality of the project.

Experiment —Testing H2: The Impact of Project Monitoring on Project Quality

To test H2, we compared the quality of two sets of virtual teams, one in which the users imposed project monitoring (referred to as *tight* monitoring), and the other one in which user project monitoring was voluntary (referred to as *loose* monitoring). For this purpose, we used a portion of the data collected as part of experiment 1. Recall that in experiment 1 we already had a set of virtual teams, namely the teams formed by MU user team and the MDI *B* developer teams, operating in voluntary project monitoring mode. We then formed another set of virtual teams by pairing each MU user team with MDI *A* teams. However, in this experiment MDI *A* teams performed the role of developers for their corresponding MU user teams (compared to the role of users they played in experiment 1). MU user teams were required to tightly monitor their projects with MDI *A* teams. This is illustrated in Figure 2(b), where MU team 1 was the user for MDI team

A1, under tight project monitoring, and was also the user for MDI team B1, under loose project monitoring (part of experiment 1). Similarly, the MU team 2 was the user for MDI team A2 and B2, and so on. Once again, each of the developer teams (i.e., MDI *A* and *B* teams) had 5 or 6 members, thus controlling for effects of team size on success of the project.

Tight and loose control was implemented as follows: In the case of virtual teams operating under imposed tight project monitoring (MU and MDI *A* teams), the developers were told to submit weekly project reports to their respective user teams. The user teams were required to review and ask for changes/actions as required, thus implementing behavioral control. In addition, MDI teams were required to conduct requirements analysis in iterative model, returning a set of intermediate artifacts which would also be reviewed and commented on by their users, thus implementing outcome control. This formed the control group in our experiment. In contrast, teams operating under voluntary user project monitoring did not have to submit regular project status reports nor any intermediate artifacts to their users. They received requirements specifications from their users, asked for clarifications where necessary, and submitted the final artifacts at the end of the project. Any communication between these teams and their users was strictly on a need-be basis. This formed the experimental group in our research design. MU teams were graded partly on the communication plans and weekly project status reports they developed for monitoring their MDI *A* teams. This ensured that MU team users spent more time and effort in monitoring their associated MDI *A* teams than MDI *B* teams. This design resulted in the two overlapping experiments 1 and 2 described previously. Table 1 illustrates the roles of MDI and MU teams in these experiments.

All student teams were formed in such a way that the technical background and average work experience of group members were almost the same across groups, thereby controlling team member heterogeneity. Table 2 provides ANOVA results comparing means of various parameters across teams. Results suggest no significant differences in the means of various

Table 1. Experimental set-up

	MU Teams	MDI *A* Teams	MDI *B* Teams	Treatment	Hypothesis Tested
Experiment 1	Users	Users	Developers	MU Users <-> MDI *B* Developers, Virtual Teams MDI *A* Users <-> MDI *B* Developers, Collocated Teams	H1
Experiment 2	Users	Developers	Developers	MU Users <-> MDI *A* Developers, Virtual and tightly controlled Teams MU Users <-> MDI *B* Developers, Virtual and loosely controlled Teams	H2

Table 2. ANOVA comparing means of variables across teams

Variable	*F*	Significance
Work Experience	0.601	0.795
Experience in Programming	1.356	0.213
Experience in Participating in Virtual Teams	0.803	0.614
Experience in Software Project Management	0.973	0.465
Experience in Systems Analysis and Design	0.543	0.841

parameters across teams confirming their homogeneity. Students had sufficient stake in the virtual team project as up to 30% of the course grade was assigned to the project.

Our research design adopts the quasi experiment approach where the participants are allotted to teams, based on certain criterion, as explained previously, and not randomly. Hence the limitations of quasi experimentation as explained in Campbell and Stanley (1966) applies to our research setting as well.

Tasks

Virtual Team Exercise

The virtual team interactions (in both experiments) were broken down into two phases: (1) *socialization,* which permitted the teams to develop relationships and negotiate communication terms and requirements; and (2) *project execution,* which allowed requirements gathering, clarifications, and exchange of analysis artifacts.

Phase 1: Socialization

It is an increasingly common practice in virtual teams to engage in formal socialization before embarking on virtual projects in order to understand each others' work styles and expectations, negotiate communications strategies and protocols, and build trust for sustained relationships (Jarvenpaa & Leidner, 1999). In our experiment, this was not feasible due to resource and other restrictions, not unlike those faced by organizations new to off-shoring as well as those involved in small, preliminary initiatives. Furthermore, our objective was to draw benchmark conclusions regarding effects of user project monitoring on teams engaged in a fully virtual team environment. Therefore we encouraged the MU and MDI teams to communicate and socialize with each other on-line before initiating actual work on the project. The virtual teams—MU, MDI *A,* and MDI *B*—socialized with each other using on-line media such as e-mail, Internet chat, bulletin boards, and e-groups for a period of 2 weeks. Project details were withheld from all teams till conclusion of the socialization phase in order to ensure that communication was more personalized and oriented towards relationship and trust building (Sarkar & Sahay, 2002) rather than requirements exchange.

Phase 2: Project Execution

Subsequent to socialization, the projects were initiated, and team roles were detailed. Marquette University has a service learning office that obtains information systems projects from nonprofit organizations and small businesses in and around Milwaukee. Such real-life projects were given to MU users. Examples of these projects include a donation management system for a nonprofit organization, a volunteer management system, an alumni website, a tracking system for battered and abused women, and a book inventory management system.

The MDI teams elicited project requirements from MU teams through various on-line media, as described previously. SSAD methodology was used in the experiment. The gathered requirements were structured using process modeling tools such as context analysis diagram (CAD), data flow diagrams (DFDs) and process specifications. MDI teams also modeled the data and associated relationships using entity relationship diagrams (ERDs). MDI teams also created screen-based prototypes as part of the requirements analysis exercise. These artifacts were submitted by the MDI teams to MU user teams as part of the deliverables.

In addition, the MDI A development teams that experienced tightly monitored projects submitted the following additional artifacts to the users:

a. A weekly status report of the project, explaining reasons for delays and plans for overcoming any slippages.
b. Any modifications to the project plan.
c. A draft (intermediate) version of all the above artifacts, midway through the project Based on their requirements, users provided feedback and corrections, which were incorporated by the developers into the final version.

Details of all these deliverables submitted by the different teams for this virtual team exercise are shown in Table 3. The table also shows several artifacts/reports that the MU teams had to submit to the course instructors.

Collocated Exercise

For the collocated team exercises, each MDI *A* team had at least one member who had prior work experience of 2 to 3 years. These individuals were asked to select an information system project they had encountered at work, to ensure realism and familiarity with system features. Each collocated team developed requirements analysis artifacts for these projects. The instructors had discussions with each group and scoped the projects such that the project complexity was almost the same as that of the virtual teams. The MDI *B* teams were asked to submit to MDI team *A* artifacts identical to those submitted to MU teams during the virtual team project (see Table 1). The entire project duration for both virtual and collocated projects was 8 weeks.

OUTCOME MEASURES

Quality of Projects

Quality of MU-MDI projects were determined through (i) expert evaluation of project artifacts produced by developer teams and (ii) user perceptions about the project deliverable quality.

Table 3. Artifacts submitted by the different teams for the virtual team projects

Artifact	**MDI *A* Teams for the Virtual Team Projects under tight Project Monitoring**	**MDI *B* Teams for both the Virtual and Collocated Projects under loose Project Monitoring**	**MU Teams for the Virtual team Projects (to be submitted to the instructors)**
Context Analysis Diagram	✓	✓	
Data Flow Diagrams	✓	✓	
Entity Relationship Diagrams	✓	✓	
Process Specifications	✓	✓	
Screen shots	✓	✓	
An intermediate version of all the above artifacts	✓		
Weekly Development Status Report	✓		
Communication Plans			✓
Risk Assessment			✓
Contingency Plans			✓
Weekly Project Status Report (to the Instructors)			(only with MDI *A* teams)
Project Closure Report			✓
Team *A* and *B* Assessment			✓

Quality of project artifacts was measured on several dimensions—namely, correctness of the artifacts (e.g., whether the data flow diagrams were drawn correctly, whether or not they satisfied user requirements), adherence of the artifacts to user requirements, and consistency of the artifacts with each other.

i. Completeness and Adherence of the Artifacts to User Requirements

Completeness and adherences were analyzed by an external expert who was not part of the MU-MDI teams. This expert had 2 to 3 years of experience in software projects and had taken courses in SSAD. The expert evaluated the completeness and adherence of each of the following artifacts:

1. Context analysis diagram
2. Data flow diagrams (DFDs)
3. Process specifications
4. Entity-relationship diagrams (ERDs)
5. Screen shots of the proposed system

The expert analyzed and scored the above artifacts for each project on a 7-point Likert-type scale. Though the expert had only 2 to 3 years of experience, by following a standard evaluation procedure such as the one outlined previously, this individual was able to arrive at an objective assessment of project quality. This evaluation was validated for consistency and accuracy by a second expert who had more than 20 years of SSAD industry experience, thus reducing possible biases in the evaluation process. The average of these scores across all artifacts for each project was taken as a measure of completeness and adherence of project artifacts to user requirements. By making the team assignments to the projects blind to the expert, we minimized subjective bias of the expert during the assessment.

ii. Consistency of the Artifacts

The expert also analyzed the consistency of the screen prototypes submitted by development teams with the DFDs and ERDs submitted. Using a 7-point Likert-type scale, the expert analyzed and scored for each project the consistency across

1. Screen prototypes and DFDs
2. Screen prototypes and ERDs

Using the same evaluation and validation procedure described in (i), an average score measuring the consistency of the project artifacts was generated.

iii. User-perceived quality

User perceptions about the quality of artifacts submitted by the developer teams were also collected through a survey questionnaire as the third measure of team performance. A 7-point Likert-type scale was used to elicit response from the user team members. Items adapted from Edwards and Sridhar (2005) are detailed in Appendix I. Scores given by all the users to a particular development team were averaged and were treated as measure of user-perceived quality. Therefore, there was one rating/score per user teams. Based on measures of quality already mentioned, hypothesis H1, which was constructed in the previous section, can be refined and are presented in Table 4.

By specifying the two dimensions of completeness and adherence as well as consistency, any errors in the assessment of the quality of the projects was thought to be minimized.

User Project Monitoring

We also measured perceived project-monitoring practices of all users and developers involved in both tight and loosely monitored projects. Responses were elicited on a 7-point Likert-type scale at the end of the project. Items are shown in Appendix I. In order to capture the responses for perceived quality and user project monitoring based on the roles they played (user/developer) and the team (collocated/virtual) with which they did the projects, different versions of the survey was prepared and administered to students at MDI and at MU.

Table 4. Detailed hypotheses based on different measures

Research Question	Hypotheses
Quality of Projects of Virtual Teams vs. Collocated Teams	**H1a:** Adherence and completeness of the requirements analysis artifacts produced by the collocated teams using face-to-face communication will be better than those produced by the virtual teams using computer-mediated communication. **H1b:** Consistency of the screen shots and requirements analysis artifacts produced by the collocated teams using face-to-face communication will be better than those produced by the virtual teams using computer-mediated communication. **H1c:** The users will perceive the quality of project artifacts produced by collocated teams using face-to-face communication to be better than those produced by the virtual teams using computer-mediated communication.
Impact of User Project Monitoring on of the Quality of Projects	**H2a:** Quality of project artifacts (as defined by the three measures of completeness & adherence, consistency, and user perception) produced by the developer teams that are closely monitored by their associated users in a virtual team mode will be better than those produce by the developer teams that were not closely monitored by their users.. **H2b:** Quality of project artifacts (as defined by the three measures of adherence & completeness, consistency, and user perception) produced by of the developers that perceived higher levels of project monitoring by their users will be better than those produced by the developer teams that perceived lower levels.

The various versions included same items for each construct but were worded differently, depending on the roles the participants played. Based on the experimental measure of perceived project management practice, hypothesis H2 can be further articulated as in Table 4.

ANALYSIS, RESULTS, AND DISCUSSIONS

A principal component analysis was performed on the items constructed for the previously mentioned measures with Varimax rotation and Kaiser normalization; the results are given in Table 5. Reliability of all these measures of (i) completeness and adherence of artifacts, (ii) consistency of project artifacts, (iii) user-perceived quality, and (iv) perceived user project monitoring practices are given in Table 6. Cronbach's alpha values of 0.70 and higher indicate construct reliability.

Performance of Collocated vs. Virtual Teams

To test hypothesis **H1,** a one-way ANOVA test was performed on the three measures of project quality, as were previously described, between virtual and collocated teams that participated in Experiment 1. Note that in this case the project artifacts are produced by the same developer teams, and the project complexity of both the virtual and collocated projects were moderated by the instructors to be almost the same. However, due to constraints in conducting the experiment, the user teams could not be the same. User project monitoring was kept loose for both virtual and collocated projects. ANOVA results are represented in Table 7.

Results indicate that all the variations (H1a, H1b and H1c) of hypothesis H1 can be rejected. Although two of the mean quality measures of collocated teams are better than that of virtual teams, they are not significantly different. This is contrary to expectations that the quality of

Table 5. Principal component analysis of various constructs indicating factor loadings of survey items

Item No	Adherence and Completeness of Project Artifacts	Consistency of Project Artifacts	User-Perceived Quality	Perceived User Project Monitoring
1	.792	.892	.882	0.663
2	.869	.885	.935	0.845
3	.699		.871	0.700
4	.400		.956	0.615
5	.680			0.759
6				0.548
7				0.686

Note. Extraction method: principal component analysis; rotation method: varimax with kaiser normalization

Table 6. Reliability coefficients (Cronbach's Alpha) of constructs

Constructs (Number of items)	Cronbach's Alpha Value
Completeness and Adherence of Project Artifacts (5)	0.70
Consistency of Project Artifacts (2)	0.71
User-Perceived Quality (4)	0.93
Perceived User Project Monitoring (7)	0.73

Table 7. ANOVA Results (Collocated vs. Virtual teams)

Construct	Mean (Collocated team)	Mean (Virtual team)	*F*-value (significance)
Completeness and Adherence of Project Artifacts	4.92	4.56	0.551(0.467)
Consistency of Project Artifacts	6.32	6.51	1.025(0.323)
User-Perceived Quality	4.91	4.75	0.616(0.435)

projects that are produced by collocated teams and that benefit from higher social presence, media-rich face-to-face communications is no better than that produced by virtual teams that use lean media. This potentially suggests that the requirements analysis phase of software projects may be successfully off-shored in full and conducted in virtual team mode without significantly affecting the quality of projects.

Effect of User Project Monitoring

To test H2a, we compared mean values of the quality measures between the tightly monitored control group and the loosely monitored experimental group. Results presented in Table 8 indicate that the completeness and adherence of project artifacts produced by the control group were significantly superior to those produced by the experimental group, suggesting that close project monitoring by users had a positive impact on this measure of project quality. However, neither the consistency of project artifacts nor the user-perceived quality differed significantly across the two sets of teams. As expected, participants in the control group perceived that their projects were indeed closely monitored, compared to those in the experimental group.

Mean values of the perceived monitoring of the virtual team were then computed. We categorized those responses that were above the mean value as *high perceived user project monitoring* and those that were below as *low perceived project monitoring*. The performance measured on all the three dimensions were then compared across these two sets, using a one-way ANOVA test. The results as presented in Table 9 indicate that artifacts produced by developers who perceived higher levels of user project monitoring practices were better on the two dimensions of completeness and adequacy, as well as user-perceived quality, as compared to those who perceived low user monitoring.

A pair-wise correlation was carried out between perceived project monitoring and the three measures of project quality, which further confirmed these findings. (These correlations in presented in table 10.)

It is important to understand the difference between imposed project monitoring as defined in the control and experimental groups and perceived project monitoring. Though ANOVA results in Table 8 indicate that the mean values of perceived project monitoring of the control group were significantly higher compared to that of the experimental group, the mean of the experimental group was significantly higher (4.35) in the Likert scale. We also observed that in the experimental group, some of the MU teams, along with their corresponding MDI *B* teams, had voluntarily adopted closer project monitoring practices. These MDI *B* teams had been submitting their project plans and intermediate artifacts to their MU user teams, thus resulting in higher levels of perceived project monitoring. From an experimental perspective, there was a positive impact of both imposed project monitoring as well as perceived proj-

Table 8. ANOVA results (tight vs. loose project monitoring)

Construct	Mean (Control group-imposed tight user project monitoring)	Mean (Experimental group—voluntary loose user project monitoring)	*F*-value (significance)
Completeness and Adherence of Project Artifacts	**5.60**	**4.50**	**4.6(0.044)**
Consistency of Project Artifacts	6.39	6.51	0.314(0.582)
User-Perceived Quality	4.61	4.75	0.076(0.785)
Perceived User Project Monitoring	**5.18**	**4.35**	**37.2 (0.000)**

Table 9. ANOVA results (perceived user project monitoring)

Construct	Mean (Perceived HIGH user project monitoring)	Mean (Perceived LOW user project monitoring)	*F*-value (significance)
Completeness and Adherence of Project Artifacts	**5.30**	**4.73**	**6.18(0.044)**
Consistency of Project Artifacts	6.41	6.49	0.107(0.768)
User-Perceived Quality	**5.01**	**4.17**	**8.91(0.003)**

Table 10. Pair-wise correlations between input and output variables

Construct	Quality of Projects		
	Completeness and Adherence of Project Artifacts (p)	***Consistency of Project Artifacts***	***User-Perceived Quality***
Perceived User Project Monitoring	**0.215 (0.021)**	0.042(0.643)	**0.281(0.002)**

Table 11. Summary of results of teams engaged in software requirements analysis

	Collocated Teams vs. Virtual Teams in Off-Shore Mode	**User Project Monitoring of Off-Shored Projects in Virtual Team Mode**	
		Control/ Experimental	**Perceived**
Completeness and Adherence of Project Artifacts	-	TPM > LPM	HUPM > LUPM
Consistency of Project Artifacts	-	-	-
User-Perceived Quality	-	-	HUPM > LUPM

Note. TPM = tight project monitoring; LPM = loose project monitoring;
HUPM = high user project monitoring; LUPM = low user project monitoring

ect monitoring on adherence of artifacts. At the same time, there was a positive impact of perceived project monitoring on user-perceived quality, possibly because of the close working relationship adopted by the users and developers. This could have occurred through informal behavioral control mechanisms such as clan control deployed by the MDI *B* teams and their corresponding MU user teams. However this issue warrants further analysis. Table 11 gives a summary of the results.

CONCLUSION

In this article we have described an exploratory study that examines two aspects of virtual teams in off-shored software development projects, specifically in the requirements analysis phase. First, we examine whether the quality of projects produced by virtual teams engaged in pure off-shore mode is at par with that of traditional, collocated teams. Secondly, within the ambit of virtual teams, we examine whether user monitoring of the projects has an impact on the quality of projects.

Contributions of the Study

Our study is one of the few to apply social presence, media richness and control theories to develop and test a research model of the antecedents of quality of software requirements analysis projects conducted in off-shore virtual team environment. As client and vendor organizations are increasingly considering off-shoring parts of requirements analysis phases, our early conclusions might enable organizations to design communications and governance structures that might facilitate virtual requirements analysis. Considering the rapid leaps in technological infrastructure globally, technology will become a moot point in this facilitation. From an academic perspective, the introduction of these two theories in an offshore context lays the foundations for extended empirical research.

We find that there is no significant difference in the quality of projects produced by virtual teams that used lean media and that by collocated teams that used rich face-to-face communications. This is similar to findings reported in Burke and Chidambaram (1999) where, despite the persistently lower social presence of leaner media, distributed groups performed better than face-to-face counterparts. Possibly, a more task-focused approach and limited social interaction may have enabled teams to generate higher quality outputs. This could be a potentially important result because it implies that off-shoring, which was so far restricted to the lower level phases of system development (such as low-level design, coding, and testing) could successfully be extended to the requirements analysis phase as well. A key benefit, of course, is that software firms could save significantly on costs by locating their business and systems analysts in off-shore locations and facilitating interactions with users through virtual channels. While this may currently be challenging, our study highlights the need for future research in improving these virtual interactions between users and off-shored development teams.

The effect of user project monitoring (control/experimental) on the quality of off-shored requirements analysis projects is ambiguous. Formal behavioral and outcome control implemented through the experimental set up had a positive effect on one measure of quality. It did not have any effect on the other two measures. Piccoli and Ives (2003) pointed out that behavior control mechanisms, which are typically used in traditional teams, have a significantly negative impact on trust in virtual teams. It was reported that behavior control mechanisms increase vigilance and create instances in which individuals perceive team members failing to uphold their obligations. On the other hand, the perceived user project monitoring had significant positive effect on two dimensions of quality (one assessed and one perceived).

We also infer that, even when project management practices were not enforced, teams might have adopted these practices to improve their performance through clan control. This observation, though anecdotal based on class observations and our analysis of perceived user project monitoring, has important implications. It provides clues that, apart from forced formal controls, informal controls existed between the users and developers when they share common goals (Choudhury & Sabherwal, 2003).

Our findings have important implications for the industry as well. Companies engaged in off-shore software development have produced strong processes around their global delivery model. However, whether the same process and project monitoring discipline will lead to success of projects conducted in pure off-shore mode in virtual team setting during the early stages of system development work has not been explored. Our research indicates that teams engaged in virtual teamwork might develop their own informal control mechanisms and even bypass the forced control mechanisms necessitated by the standard operating procedures while doing their projects. The firms (viz. both the clients and software developers) engaged in off-shore work should develop a conducive climate for team members to develop these informal controls that seem to affect project quality. Apart from this, our study fills the gap in the literature in the area of analysis of quality of projects implemented by virtual teams engaged in off-shore system requirements analysis. Further research is needed to confirm our exploratory findings.

Limitations of the Study: Opportunities for Future Research

Use of Experiments

Literature in the area of virtual teams has mainly followed three research methodologies—case studies, industry surveys, and experiments. Experimental methods make possible the careful observation and precise manipulation of independent variables, allowing for greater certainty with respect to cause and effect, while holding constant other variables that would normally be associated with it in field settings (Damian et al., 2000). They also encourage the investigator to try out novel conditions and strategies in a safe and exploratory environment before implementing them in the real world (McGrath, 1984). The industry is yet to adopt off-shoring of the requirements analysis phase. This precludes the use of case study or industry survey for this research. Hence, we used experiments where we can explore this emerging phenomena.

In our experiment, MDI *A* teams played the roles of both users (in Experiment 1) and developers (in Experiment 2). The dual roles could have created conflicts that might have affected (positively or negatively) their project quality. The same is true with MDI *B* teams, who performed the roles of consultants for both MU teams as well as MDI *A* teams. MU teams also had to manage two projects: one with tight monitoring (with MDI *A* teams) and the other with loose monitoring (with MDI *B* teams). To remove the confounding effects of dual roles played by the teams, it is recommended that a true controlled factorial experiment be conducted to verify our findings.

Use of Students as Surrogates

There are criticisms for the use of students in academic experiments as surrogates. However, MBA students have been used as surrogate users in a range experiments conducted (see, e.g., Briggs, Balthazard, & Dennis, 1996; Hazari, 2005). Even in requirement negotiation phase, students with work experience were taken as users for developing a small system (Damian et al., 2000). Remus (1986) argued that graduate students could be used as surrogates for managers in experiments on business decision making. Students often represent a typical working professional and organizational member due to the variety of backgrounds and goals (Dipboye & Flanagan, 1979). Studies in industrial organization psychology and organization behavior have found that results obtained from students were similar to those from managers (see, e.g., Locke, 1986). Despite the fact that users and developers in our experiments had 2 to 4 years of work experience, limitations of using students as surrogates are still applicable in our study. As the industry evolves, we suggest the extension of these experiments to business settings.

Complexity of Projects

Requirements analysis is intensive, and hence it is not possible to completely replicate in student experiments. However, our objective was to study the research questions on comparable, relatively well-defined small projects in which complexity of requirements analysis is not high.

Though the experiments were carefully designed, the projects were limited in scope and size compared to large-scale industrial projects. Furthermore, no formal measures of complexity were used in the study so that we could compare the projects used in the experiments with real-world industrial projects. Further research is needed to assess the impact of these findings on large-scale industrial projects with complex requirements.

Future Research Directions

One way of dealing with the lack of realism in laboratory experiments is to use multiple methods (McGrath, 1984) so that strengths of some compensate weaknesses of others. To truly test the predictive ability of the research results, the studies must also involve a multiplicity of research methodologies in order to avoid biases due to the methods used (Jarvenpaa, Knoll, & Leidner, 1988). Simulated laboratory negotiations could be complemented by field studies or validations (whose strength is realism), if the lack of realism is an issue. In our research, internal validity of results was established through conducting experiments in a controlled environment. We expect to conduct external validity through industry survey.

Finally, while we have explored one variable of project control, quality of projects can be affected by other variables such as team motivation, trust, cohesion, coordination, and communication (Chidambaram, 1996; Jarvenpaa et al., 1998; Lurey & Raisinghani, 2001). Hence, a comprehensive model that defines all factors affecting the quality of off-shored software requirements analysis projects must be developed. Further research is required to determine how informal controls develop between the virtual team members. One cause may be the amount of initial online socialization, when the teams familiarize with each other before the start of the project, for the design of such experiments in the future. Since it may not always be feasible to make experimental and control groups adhere to experimental requirements in a classroom setting, a flexible approach is needed in experimental design.

REFERENCES

Andres, P. (2002). A comparison of face-to-face and virtual software development teams. *Team Performance Management, 8*(1/2), 39-48.

Archer, N. P. (1990). A comparison of computer conferences with face-to-face meetings for small group business decisions. *Behavior & Information Technology, 9*(4), 307-317.

Battin, R., Crocker, R., Kreidler, J., & Subramanian, K. (2001). Leveraging Resources in global software development. *IEEE Software, 18*(2), 70-77.

Briggs, R. O., Balthazard, P. A., & Dennis, A. R. (1996). Graduate business students as surrogates in the evaluation of technology. *Journal of End User Computing, 8*(4), 11-17.

Burke, K., & Aytes, K. (1998). A longitudinal analysis of the effects of media richness on cohesion development and process satisfaction in computer-supported workgroups. In *Proceedings of the 31st Hawaii International Conference on Systems Sciences* (pp. 135-144).

Burke, K., & Chidambaram, L. (1996). How much bandwidth is enough? A longitudinal examination of media characteristics and media outcomes. *MIS Quarterly, 23*(4), 557-580.

Campbell, D. T., & Stanley, J. C. (1966). *Experimental and quasi-experimental designs for research*. Chicago: Rand McNally.

Carmel, E. (2006). Building your information systems from the other side of the world: How Infosys manages time zone differences. *MIS Quarterly Executive, 5*(1), 43-53.

Chidambaram, L. (1996). Relational development in computer-supported groups. *MIS Quarterly, 20*(2), 143-163.

Chidambaram, L., & Bostrom, R. (1993). Evolution of group performance over time: A repeated measures study of GDSS effects. *Journal of Organizational Computing*, *3*(4), 443-469.

Choudhury, V., & Sabherwal, R. (2003). Portfolios of control in outsourced software development projects. *Information Systems Research*, *14*(3), 291-314.

Crisp, C. B. (2003). Control enactment in global virtual teams. *Dissertation Abstracts International.* (UMI No.)

Damian, D. E, Eberlein, A., Shaw, M. L. G., & Gaines, B. R. (2000). Using different communication media in requirements negotiation. *IEEE Software*, *17*(3), 28-36.

Damian, D. E., & Zowghi, D. (2003). An insight into interplay between culture, conflict and distance in globally distributed requirement negotiations. In *Proceedings of the 36th Hawaii International Conference on System Sciences*.

Dipboye, R. L., & Flanagan, M. F. (1979). Research setting in industrial and organization psychology: Are findings in the field more generalizable than in laboratory. *American Psychologist, 34*(2), 141-150.

Ebert, C., & De Neve, P. (2001). Surviving global software development. *IEEE Software, 18*(2), 62-69.

Edwards, K., & Sridhar, V. (2005). Analysis of software requirements engineering exercises in a global virtual team setup. *Journal of Global Information Management*, *13*(2), 21-41.

Favela, J., & Pena-Mora, F. (2001). An experience in collaborative software engineering education. *IEEE Software, 18*(2), 47-53.

Foster, S., & Franz, C. (1999). User involvement in information systems development: A comparison of analyst and user perceptions of system acceptance. *Journal of Engineering Technology Management, 16*(3-4), 329-348.

Hartwick, J., & Barki, H. (1994). Explaining the role of user participation in information system use. *Management Science, 40*(4), 440-465.

Hazari, S. I. (2005). Perceptions of end-users on the requirements in personal firewall software: An exploratory study. *Journal of Organizational and End User Computing*, *17*(3), 47-65.

Hoffer, J., George, J., & Valacich, J. (1999). *Modern systems analysis and design*. Reading, MA: Addison Wesley.

Jarvenpaa, S., Knoll, K., & Leidner, D. (1998). Is anybody out there? Antecedents of trust in global virtual teams. *Journal of Management Information Systems, 14*(4), 29-64.

Jarvenpaa, S., & Leidner, D. (1999). Communication and trust in global virtual teams. *Organization Science, 10*(6), 791-815.

Kircsh, L. J. (1996). The management of complex tasks in organizations: Controlling the systems development process. *Organizational Science, 7*(1), 1-21.

Kircsh, L., Sambamurthy, V., Ko, D., & Purvis, R. (2002). Controlling information systems development projects: The view from the client. *Management Science*, *48*(4), 484-498.

Lin, W., & Shao, B. (2000). A relationship between user participation and system success: A simultaneous contingency approach. *Information & Management, 37*(6), 283-295.

Lind, M. (1999). The gender impact of temporary virtual work groups. *IEEE Transactions on Professional Communication*, *42*(4), 276-285.

Locke, E. A. (1986). *Generalizing from laboratory to field setting: Research finding from industrial organization, organization behavior, and human resource management.* Lexington, MA: Lexington Books.

Lurey, J., & Raisinhgani, M. (2001). An Empirical study of best practices in virtual teams. *Information & Management, 38*(8), 523-544.

McDonough, E., Kahn, K., & Barczak, G. (2001). An investigation of the use of global, virtual, and collocated new product development teams. *The Journal of Product Innovation Management*, *18*(2), 110-120.

McGrath, J. (1984). *Groups: Interaction and performance*. Upper Saddle River, NJ: Prentice Hall.

National Association of Software and Service Companies. (2005). *Indian IT industry*. Retrieved March 3, 2005, from http://www.nasscom.org/

Piccoli, G., & Ives, B. (2003). Trust and the unintended effects of behavior control in virtual teams. *MIS Quarterly*, *27*(3), 365-395.

Powell, A., Piccoli, G., & Ives, B. (2004). Virtual teams: A review of current literature and directions for future research. *The DATABASE for Advances in Information Systems, 35*(1), 6-36.

Remus, W. E. (1986). An empirical test of the use of graduate students as surrogates for managers in experiments on business decision making. *Journal of Business Research, 14*(1), 19-25.

Rice, R. E., & Love, G. (1987). Electronic emotion: Socio-emotional content in a computer-mediated communication network. *Communication Research*, 14(1), 85-108.

Sharda, R., Barr, S. H., & McDonnell, J. C. (1988). Decision support system effectiveness: A review and an empirical test. *Management Science, 34*(2), 139-157.

Schimdt, J. B., Montoya-Weiss, M. M., & Massey, A. P. (2001). New product development decision-making effectiveness: Comparing individuals, face-to-face teams and virtual teams. *Decisions Sciences*, *32*(4), 575-600.

Sarkar, S., & Sahay, S. (2002). Information systems development by US-Norwegian virtual teams: Implications of time and space. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences* (pp. 1-10).

Short, J., Williams, E., & Christie, B. (1976). *The social psychology of telecommunications*. London: Wiley.

Sridhar, V., Nath, D., & Malik, A. (in press). Analysis of user involvement and participation on the quality of IS planning projects: An exploratory study. *Journal of Organizational and End User Computing*.

Stevenson, W., & McGrath, E. W. (2004). Differences between on-site and off-site teams: Manager perceptions. *Team Performance Management, 10*(5/6), 127-132.

Townsend, A., DeMarie, A. M., & Hendrickson, A. R. (1998). Virtual teams: Technology and the workplace of the future. *Academy of Management Executive*, *12*(3), 17-29.

Walther, J. (1995). Relational aspects of computer-mediated communication: Experimental observations over time. *Organization Science*, *6*(2), 186-203.

APPENDIX I

Survey Questionnaire Items

User-Perceived Quality

1. The screen prototypes submitted by the remote team adequately reflected the requirements conveyed by my local team.
2. My local team has been satisfied with the quality of the deliverables submitted by the remote team.
3. My local team found final deliverables to be free of errors.
4. The final artifacts submitted by my remote team adequately reflected the requirements conveyed by my local team.

Project Monitoring

1. My local team tracked the project's progress closely.
2. The remote team submitted the detailed project plan on time.
3. My local team suggested changes to the initial project plan submitted by the remote team.
4. Any changes in the project plan were communicated to us immediately by the remote team.
5. My local team regularly monitored the progress of the project.
6. My remote team closely adhered to the submitted project plan.
7. My remote team regularly submitted weekly status reports.

Dhruv Nath is professor and chairman of the doctoral program at the Management Development Institute (MDI), Gurgaon, India. Earlier, he was senior vice president at NIIT Ltd, a leading software and education company. Prof. Nath holds a BTech in electrical engineering and a PhD in the area of computer science, both from IIT Delhi. He has conducted Top Management Workshops in IT and has been a consultant in the area to several organizations such as Glaxo, Unilever, Gillette, Nestle, etc. He has published research papers in international journals such as the IEEE transactions on Computers*, and has written two books on managing IT.*

Varadharajan Sridhar is Professor in Information Management at the Management Development Institute, India. He received his PhD from the University of Iowa, USA. Dr. Sridhar's primary research interests are in the area of telecommunication management and policy and Global Software Development. He has published his research work in European Journal of Operational Research, Telecommunication Systems, International Journal of Business Data Communications and Networking, Applied Econometrics and International Development, Information Resource Management Journal, Journal of Global Information Management, Journal of Regional Analysis and Policy, and Journal of Information System Security. *He was the recipient of the Nokia Visiting Fellowship awarded by the Nokia Research Foundation. He is an associate editor of the* International Journal of Business Data Communications and Networking *and is on the editorial board of the* Journal of Global Information Management.

Monica Adya received her PhD from Case Western Reserve University. She researches in the area of knowledge management systems, particularly for business forecasting. She also conducts research on IT workforce issues and virtual team management. Her work appears in Human Resource Management, Journal of Managerial Psychology, Information Systems Research, Information Technology & People, International Journal of Forecasting, and Journal of Forecasting *among others. She was recently a co-recipient of a grant from 3M Foundation for examining the impact of IT outsourcing on American education.*

Amit Malik is a doctoral candidate in information management at the Management Development Institute, India. He has received bachelor's degree in computer engineering from National Institute of Technology, India. He has over 3 years of professional experience in software design and development, process designing and implementation. His research interests include virtual teams, software development off-shoring, and knowledge management.