2007

Investigating an ‘Agile-Rigid’ Approach in Globally Distributed Requirements Analysis

Vanita Yadav
Management Development Institute, India, vanitay@gmail.com

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

Dhruv Nath
Management Development Institute, India, dhruv@mdi.ac.in

V Sridhar
Management Development Institute, India, sridhar@mdi.ac.in

Follow this and additional works at: http://aisel.aisnet.org/pacis2007

Recommended Citation
http://aisel.aisnet.org/pacis2007/12

This material is brought to you by the Pacific Asia Conference on Information Systems (PACIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in PACIS 2007 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.
10. Investigating an ‘Agile-Rigid’ Approach in Globally Distributed Requirements Analysis

Vanita Yadav  
Management Development Institute, India  
vanitay@gmail.com

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

Dhruv Nath  
Management Development Institute, India  
dhruv@mdi.ac.in

V. Sridhar  
Management Development Institute, India  
sridhar@mdi.ac.in

Abstract
The global software development environment brings with itself abundant business opportunities as well as challenges in terms of coordination, communication and control. Recent years have also witnessed the growth of the agile movement. To address the global software development challenges there is a need to combine the flexibility offered by the growing agile development approaches with the rigidity offered by the traditional plan-based approaches. This paper reports an exploratory quasi-experimental study, which investigates the performance of requirements analysis projects in an ‘agile-rigid’ distributed environment. The study yields several interesting conclusions that can assist organizations in managing their global software projects more effectively. Our experiment indicates that project monitoring and control, project communication, and process facilitation between peer teams significantly influence the success of such projects. Creation of an agile-rigid environment can help organizations mitigate various risks inherent in globally distributed software development.

Keywords: Globally distributed software development, requirements analysis, agile development, offshoring, project management

Introduction
The forces of globalization have today made software development a multi-site, multicultural and globally distributed activity. There is a growing body of literature in management that highlights the globalization phenomenon and global corporations’ need to plan for such work environments (Yadav et al. 2007). Software development has increasingly moved away from the traditional co-located model, often called on-site development, to the offshoring model (Edwards and Sridhar 2006) in which global virtual teams collaborate across national borders (Carmel 1999). The development of software in globally distributed projects is one of today’s big challenges (Sangwan et al. 2007).

Global software development is defined as software development that uses teams from multiple geographic locations (Sangwan et al. 2007). These could be in terms of offshore sourcing of software development work (Carmel and Tijia 2005), to an in-house offshore facility in another country or to a third party service provider located in another country. There is a growing debate in literature on what can be and what cannot be offshored and carried out in a globally distributed mode. There are some researchers who state that certain activities, like coding, are a better fit for offshore locations while other activities, such as requirements gathering, are better to be carried out onshore within the client’s country.
(Carmel and Tijia 2005). However, as organizations become more 'virtual, distributed development will become more apparent throughout the entire life-cycle, including analysis of software requirements in the early stages of the development lifecycle (Evaristo et al. 2005). There is a growing stream of researchers (Bhat et al. 2006; Edwards et al. 2005; Edwards et al. 2006; Nath et al. 2006; Ocker et al. 1995; Damian et al. 2000; Evaristo et al. 2005) who argue in favor of distributed requirements engineering. Requirements engineering is a software project’s most difficult phase (Hofmann and Lehner 2001) and the success of this phase is vital for project success. Despite the abundant literature available on globally distributed virtual teams (Powell, Piccoli and Ives 2004) and IT outsourcing (Dibbern et al. 2004), there are very few studies addressing the critical requirements analysis phase in distributed software development. This can be attributed to the relative newness of this area or to a common belief of offshoring only mechanical phases like coding.

Recent years have witnessed the growth of the agile movement. Continued dissatisfaction with the traditional plan-driven (heavyweight) methodologies has led to the introduction of various agile (lightweight) methodologies, like eXtreme programming, Scrum, Crystal etc. (Fruhling and Vreede 2006; Lindstrom and Jeffries 2004). Agile development methodologies provide organizations with an ability to rapidly evolve Information Systems (IS) solutions (Meso and Jain 2006). The global software development environment brings with itself abundant business opportunities as well as challenges in terms of coordination, communication, culture and technology. To address these challenges many researchers propose that firms must have ambidextrous capabilities (Lee, Delone and Espinosa 2006) and combine the flexibility offered by the growing agile development approach with the traditional plan-based approach (Lee et al. 2006; Agerfalk et al. 2006; Ramesh et al. 2006). Further, the there is paucity of literature available from this fresh perspective. Agerfaulk et al. (2006) point out that practice is ahead of research in the globally distributed agile area and there is an evident need to better conceptualize and theorize fundamental underpinnings. Additionally, the critical requirements phase of globally distributed software projects is yet to be empirically explored by IS researchers.

The aim of this research is to address this gap in IS literature. The objective of this exploratory study is to model factors and relationships that affect project success in an agile-rigid framework during the requirements analysis phase of global software development. We adopt quasi-experimentation methodology in an academic setting. The paper proceeds as follows. In the next section, we discuss the main concepts of the paper by developing an agile-rigid perspective. We then move towards development of our conceptual model and posit our research hypotheses. We then describe the methodology adopted for research and discuss the academic quasi-experiment. We conclude with our research findings, contributions and directions for future research.

**Developing an ‘Agile-Rigid’ Perspective in Global Software Development**

**Global Software Development (GSD)**

We adopt the viewpoint of Sangwan et al. (2007) in defining Globally Distributed Software Development or Global Software Development (GSD) as “software development that uses teams from different multiple geographic locations from different countries”. These teams can be from the same organization or they may be collaborations that involve different organizations. Carmel et al. (2005) call this as “offshore sourcing”, which includes both offshore outsourcing to a third-party provider as well as offshore insourcing to an internal group within a global corporation.
Many researchers have highlighted the potential benefits from GSD (Carmel 1999; Agerfalk and Fitzgerald 2006; Sangwan et al. 2007). These include reduced development costs, reduced cycle time, cross-site modularization of development work, access to larger and better skilled developer pool, shared best practices and closer proximity to customers. Many success stories of GSD are available in literature (Ebert and Neve 2001; Battin et al. 2001; Carmel 2006). However, researchers also emphasize that GSD brings with it many pros and cons and if not managed carefully can turn any GSD venture into a loss making enterprise (Sangwan et al. 2007). Agerfalk et al. (2006) argue that while geographic distance by itself many induce a number of problems; increased geographic distance also often increases temporal and sociocultural distance. When people are not co-located (different geography) they often rely on asynchronous communication like email or synchronous communication like chat, teleconferencing or videoconferencing. People also work in different time zones which results in the emergence of temporal distance. Additionally, GSD involves collaboration of people from different countries with different cultural and social backgrounds. This leads to the surfaced of sociocultural distance.

Requirements Analysis in GSD

One of the hardest parts of system’s development is deciding ‘what the system should do’, that is in determining the system requirements (Crowton and Kammerer 1998, pp. 227). According to Byrd, Cossick and Zmud (1992), requirements analysis typically involves an analyst working with end users to establish an understanding of organizational information processing needs; developing IS objectives; designing and evaluating IS alternatives; communicating the results to superiors and end users; and performing a systems audit. Hoffer, George and Valacich (2005) describe analysis as a large and involved process, which is divided into three sub-phases:

- **Requirements Determination**: in this sub-phase the analysts work with the users to determine what the users want from a proposed system.
- **Requirements Structuring**: This activity creates a thorough and clear description of current business operations and new information processing services. This involves modeling the requirements.
- **Alternative Generation and Selection**: This sub-phase results in choice among alternative strategies for subsequent systems design.

We adopt the view of Hoffer, George and Valacich (2005), and focus on the first two sub-phases of requirements analysis- requirements determination and requirements structuring. Globally distributed requirements analysis generally includes a team of analysts and users working together using technologies like computer-mediated conferencing, instant messaging, email, teleconferencing, web based group support systems like wiki’s. Examples of artifacts of the requirements analysis phase include drawing context diagrams, modeling the requirements, creating prototypes, creating a data dictionary, prioritizing requirements and allocation requirements to sub-systems (Weigers 2003).

As organizations become more global and stakeholders more distributed, getting the requirements right will pose a greater challenge. Nowadays, organizations are offered a sophisticated array of multimedia meeting systems with video, audio and computer support for remotely specifying requirements (Damian et al. 2000). In spite of the growing literature on GSD there are a very few research papers that address the requirements analysis phase.
This can be ascribed to the relative newness of this area and to the unfamiliarity of most software developers with the phenomenon of distributed group dynamics (Boehm, Grunbacher and Briggs 2001).

**Agile Software Development**
Agile software development advocates frequent and regular software releases (Hunt 2006). Agile methods focus on producing a working solution in conjunction with changing user requirements. Traditional development approaches also try to develop working solutions, but the difference lies in the focus on ‘changing user requirements’. Agile approaches focus on fast deliverables, dynamic management of requirements, and fast iterations and incrementations (Fruhling and Vreede 2006). Although these new approaches have been proposed and many positive benefits speculated, there have been very few empirical field studies on operationalizing results (Fruhling and Vreede 2006) and developing theories in this area (Agerfalk and Fitzgerald 2006). The fundamental differences between agile and traditional software development adapted from Nerur et al. (2005) are highlighted in Table 1.

<table>
<thead>
<tr>
<th>Fundamental Assumptions</th>
<th>Traditional</th>
<th>Agile</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Systems are fully specifiable, predictable and can be built through extensive planning</td>
<td>High quality adaptive software can be developed by small teams using the principles of continuous design improvement and testing based upon rapid feedback and change</td>
</tr>
<tr>
<td>Control</td>
<td>Process-centric</td>
<td>People-centric</td>
</tr>
<tr>
<td>Management</td>
<td>Command-and-control</td>
<td>Leadership-and-collaboration</td>
</tr>
<tr>
<td>Role Assignment</td>
<td>Individuals- favors specialization</td>
<td>Self-organizing teams- encouraging role interchangeability</td>
</tr>
<tr>
<td>Communication</td>
<td>Formal</td>
<td>Informal</td>
</tr>
<tr>
<td>User’s Role</td>
<td>Important</td>
<td>Critical</td>
</tr>
<tr>
<td>Development Model</td>
<td>Lifecycle model (waterfall, spiral or some variation)</td>
<td>Evolutionary delivery model</td>
</tr>
</tbody>
</table>

**Challenges of Using Agile Methods in GSD**
Temporal, geographical, and sociocultural distance in GSD create complexities in realizing the key concepts of agile methods. For instance, pair programming, on-site customer collaboration, and face-to-face interaction is severely cut down in GSD which can negatively influence the way in which agile methods are implemented. Also, given the risks inherent in GSD, the natural tendency is probably to favor plan-based approaches (Agerfalk and Fitzgerald 2006). Although there has been some preliminary research on how to apply XP in GSD (Nago-The et al. 2005), a common belief is that agile methods are not applicable in GSD. Some emphasize that there is a need of an increased understanding of the characteristics of the agile methods and how it can be applied to reduce the negative influence of distance (Holmstrom et al. 2006). Despite the evidence of successful agile software development, its application in GSD has yet to gain momentum (Holmstrom et al. 2006). There are very few recent studies that address the combination of agility and GSD.

**Research Model**
Meso and Jain (2006) argue that organizations that use heavyweight methodologies need to reconsider agile practices and integrate them as suitable. This research acknowledges the views of researchers arguing in favor of blending the agile and traditional plan-based methodologies in GSD. This research aims to go a step further and analyzes the requirements analysis phase of agile-rigid GSD. The key characteristics of the agile-rigid GSD
environment adapted from a current stream of research (Holmstrom et al. 2006; Lee, Delone and Espinosa 2006; Ramesh et al. 2006; Lee et al. 2006) is outlined in Table 2.

**Table 6: Characteristics of the ‘agile-rigid’ GSD Environment (Requirements Analysis)**

| 1. Size: Small teams of less than 10 members |
| 2. Project Management: blend of planned and flexible approaches |
| 3. Technology: High usage of web/Internet-based tools |
| 4. Artifacts: improvement of artifacts through feedback and iterative development. Development of prototypes (screen-based or small working solutions) for quicker feedback and development. |
| 5. Standard software development methodology (e.g., object-oriented analysis) |

Based upon the attributes in Table 2 we posit our research model (see Figure 1) which guides our hypothesis development process. Addressing the GSD challenges of control and communication three predictor variables—project monitoring and control, project communication and process facilitation by a coordinator were identified for this study. The impact of these predictor variables on perceived project success is examined in an agile-rigid GSD environment.

![Research Model](image)

**Figure 1: Research Model**

The agile-rigid’ GSD environment offers rigor of the traditional approaches and flexibility of the agile approaches within the teams. Rigor is provided by incorporating formal structures of the traditional approach, such as, development of a project plan, communication plan and project status tracking. The ‘rigid’ aspect of this environment is captured by the ‘project monitoring and control’ and ‘project communication’ variables. Agility is allowed through simplified project planning. Elaborate project management techniques are tailored to make them ‘light-weight’. ‘Simple rules’ (Augustine et al. 2005) are adopted. Formal as well as informal channels of communication are encouraged. Formal channel includes communication between the analysts and users through an appointed coordinator. Informal channels include communication between analysts and users using informal web-based technologies like e-groups. Additionally, it adopts the agile philosophy of iterative development and embraces requirements change. We adapted items from literature to measure predictor and outcome variables for this study.

**Outcome Variable—Project Success**

GSD mainly involves the usage of globally distributed virtual teams. Although several researchers have compared the performance of traditional co-located teams and virtual teams,
the conclusions of these efforts have been mixed. A majority of the early work has detected no difference between the two types of teams (Burke and Aytes 1998). Similar to the more generic “performance” measure, most researchers have found no significant differences between traditional teams and virtual teams when examining decision quality (Chidambaram and Bostrom 1993). In this study we measure the performance of the GSD project by the team members’ perception on project success. Researchers have identified three dimensions of project success- client satisfaction, perceived quality of the project and success with the implementation process (Mahaney and Lederer 2006). Based on these three dimensions Mahaney and Lederer (2006) developed a comprehensive instrument for measuring Information Systems project success and we refer to the instrument developed in this study in our work.

**Predictor Variable- Process Facilitation**

Anson et al. (1995) and Miranda et al. (1999) suggested process facilitation as a structure that assisted in creation of productive meeting processes. Process facilitation is defined as the provision of procedural structure and general support to groups (Eden 1990; Miranda and Bostrom 1999). Ramesh et al. (2006) reported that a primary point of contact for each location in GSD helped in facilitating communication across the teams. Hence we posit:

**H1a:** Teams provided structure by a coordinator (process facilitation) will achieve significantly better communication effectiveness than teams without any coordinator in an agile-rigid GSD environment

Lee, Delone and Espinosa (2006) indicate that assigning ‘point persons’ in offshore software development plays pivotal role in sensing and responding to emergent problems on a real time basis. We posit that process facilitation through a coordinator in GSD teams will result in better outcomes. Process facilitation was introduced as a treatment in our study. We measured process facilitation using a dichotomous variable indicating presence and absence of a coordinator. We also posit that:

**H1b:** Teams provided structure by a coordinator (process facilitation) will achieve significantly better perceived project success than teams without any coordinator in an agile-rigid GSD environment

**Predictor Variable- Project Monitoring and Control**

Project monitoring involves tracking, interpreting and transmitting status information (Crisp 2003). Crisp (2003, pp. 11) in his dissertation on control enactment in global virtual teams defines control as “attempts to influence members of a collective to engage in behaviors that lead to the attainment of collective goals”. Crisp (2003) reports that control can be implemented through processes and structures (e.g., deadlines). Control theory suggests that controllers utilize two modes of formal control: behavior and outcome (Eisenhardt 1985). In behavior control, controllers define appropriate steps and procedures for task performance and evaluate controlees’ performance according to their adherence to the prescribed procedures (Kirsch et al. 2002). In outcome control, specific desired task outputs are described in which controllers define appropriate targets and allow controlees to decide how to meet those output targets. Performance evaluation then focuses upon the extent to which targets were met, and not on the processes used to achieve the targets (Kirsch et al. 2002). This study focuses on measuring monitoring and control (behavior and outcome) and posits that:

**H2a:** Increased project monitoring and control between remote team members lead to better perceived project success in an agile-rigid GSD environment
H2b: Increased project monitoring and control between remote team members lead to better communication effectiveness in an agile-rigid GSD environment

**Predictor Variable- Project Communication**

Hulnick (2000, pp. 33) noted that “if technology is the foundation of the virtual business relationship, communication is the cement”. The virtual environment presents considerable challenges to effective communication including time delays in sending feedback, lack of a common frame of reference for all members, differences in salience and interpretation of written text, and assurance of participation from remote team members (Crampton 2001).

H3a: Increased project communication between remote team members lead to better communication effectiveness in an agile-rigid GSD environment

Delisle (2001) in her dissertation on success and communication in virtual project teams raises a pertinent question “how do virtual project teams communicate and divide the type of work needed to complete a project successfully?” She argues that although the organization may decide the team’s protocol in a rules-based manner, the actual functioning of the virtual project teams appears more evolutionary. GSD virtual project teams face geographic as well as time zone separation. There is a need to plan for who needs what information, when they will need it and how it would be given to them. Hence we posit that:

H3b: Increased project communication between remote team members lead to better perceived project success in an agile-rigid GSD environment

**Mediating Variable- Communication Effectiveness**

Sproull and Kiesler (1986) point out that technology tends to restrict the communication process because electronic media are intrinsically leaner than face-to-face communication and convey a limited set of communication cues. Thus, teams operating in the virtual environment face greater obstacles to orderly and efficient information exchange than their counterparts in the traditional context, a difficulty that is compounded when the virtual team is global in nature. Paul et al. (2005) defined items for perceived participation and communication in the context of collaborative conflict management exercises. Piccoli et al. (2004) analyzed team member communication on the effectiveness of virtual teams and indicated that the most satisfied team members were in virtual teams with effective coordination and communication. They hypothesized that communication effectiveness mediates the relationship between team control structure and team effectiveness. We also posit that communication effectiveness partially mediates the relationship between project processes (project control and project communication) and project success. We refer to the instrument of Piccoli et al. (2004) and Paul et al. (2005) in our work to measure communication effectiveness. Thus we posit:

H4: Communication effectiveness partially mediates the relationship between project processes (project monitoring & control and project communication) and perceived project success in an agile-rigid GSD environment

**Research Methodology**

**Quasi-experimentation**

A large amount of published research in IS is based around some form of laboratory experiment (Introna and Whitley 2000). Faculties in many universities and business schools have set up distributed software engineering laboratories for conducting virtual team exercises in their courses (Edwards and Sridhar 2006). Powell et al. (2004) have listed a number of studies involving students in global virtual teams. The methodology adopted for
this research is also of such a kind. We conducted a quasi-experiment in an academic setting involving students from two countries.

Defining quasi-experimental designs, Campbell and Stanley (1966, pp. 34) state that “there are many natural settings in which a research person can introduce something like experimental design into his scheduling of data collection procedures (e.g., the when and to whom of measurement), even though he lacks the full control over the scheduling of experimental stimuli (the when and to whom of exposures) which makes a true experiment possible.” We implemented the post-test only non-equivalent control group quasi-experiment design of the following form:

\[
\begin{array}{ccc}
M & X & O_1 \\
M & & O_2 \\
\end{array}
\]

‘M’ stands for matching (a priori equalization of the two groups for the factors that have to be controlled), ‘X’ stands for treatment and ‘O’ stands for observation or measurement. The groups were controlled in terms of team size, requirement artifacts, technology usage (e-group) and standard software development methodology (object-oriented analysis). The treatment was that one set of team had an appointed coordinator and the other set did not have a coordinator. Our unit of analysis was at the individual level. We measured the effect of predictor variables on the outcome variable as perceived by the individual team members.

**GSD Project**

The purpose of this exploratory quasi-experiment was to engage students in globally distributed requirements analysis. The user team comprised of students from Marquette University (MU), US and the analyst team comprised of students from Management Development Institute (MDI), India. The user team in US was enrolled for a course in IT Project Management. The analyst team in India was enrolled for Management Information Systems (MIS) course having comprehensive coverage of systems analysis and design.

Each team of users was paired with a team of analysts (see Table 3). They worked together using web-based communication technology mainly e-groups (Google groups). 16 Google groups were formed for the GSD project and user-analyst teams were added to each Google group. Each user team was provided a set of requirements and the analyst team was required to develop requirements artifacts and screen-based prototypes of a business information system requested by the user team (see Table 4). To facilitate ‘changing requirements’ in GSD a brief set of requirements for each project was given to the MU users at the onset of the project. The analysts developed the first iteration artifacts based on these requirements. A second set of more detailed requirements were then given to the MU user team by the course instructor and the analysts then created the second iteration incorporating the changed set of requirements.

<table>
<thead>
<tr>
<th>Treatment: Coordinator Present</th>
<th>Treatment: Coordinator Absent</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Setting:</strong> agile-rigid GSD (requirements analysis)</td>
<td></td>
</tr>
<tr>
<td>Total Sample Size= 102</td>
<td></td>
</tr>
<tr>
<td>Treatment Sample Size= 53 Analysts=26, Users=27 (8 pairs of user-analyst teams. 8 teams of 3-4 analysts and 8 teams of 2-3 users)</td>
<td></td>
</tr>
<tr>
<td>Treatment Sample Size= 49 Analysts=26, Users=23 (8 pairs of user-analyst teams. 8 teams of 3-4 analysts and 8 teams of 2-3 users)</td>
<td></td>
</tr>
</tbody>
</table>

The e-groups technology served as a one-point meeting ground for the users and analysts. It added transparency to the activities and discussions between the users and analysts as
messages and artifacts posted on the group were readily visible to all the e-group members. Every e-group member could post messages, files or start a discussion on the e-group. All the members posted their personal details on the e-groups along with their photographs at the start of the project to get acquainted with each other. This was followed by the requirements determination and requirements modeling. Real life projects at Marquette University were used to create the simulated GSD projects. This enabled the projects to closely mirror a real business environment.

### Table 8: GSD Project Artifacts/ Deliverables

<table>
<thead>
<tr>
<th>MDI Project Deliverables</th>
<th>MU Project Deliverables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simplified Project Management Plan (to MU users with a copy to MDI instructor).</td>
<td>Simplified Project Charter, Project Schedules and Resource Allocation and Communication Plans (to the MU Instructor).</td>
</tr>
<tr>
<td>No formal resource allocation at the analyst end (autonomous analyst teams).</td>
<td></td>
</tr>
<tr>
<td>Simplified Project Status Reports to MU Users</td>
<td>Simplified Risk Assessment and Contingency Plans (to the MU Instructor)</td>
</tr>
<tr>
<td>First Iteration (Vision document, Use Case Model, Supplementary Specifications and Screen-based Prototypes of the business IS)</td>
<td></td>
</tr>
</tbody>
</table>

**Changes in User Requirements**

| Second Iteration (Vision document, Use Case Model, Supplementary Specifications and Screen-based Prototypes of the business IS) | Simplified Project Status Report and Project Closure Report (to the MU Instructor) |

### Analysis

A survey instrument was used to collect data at the end of the quasi-experiment. Survey responses for 22 items were generated from 102 respondents (users and analysts). All items were measured on a seven point Likert-type scale, where one indicated strong disagreement and seven indicated strong agreement with the construct item. Demographic data was also collected through direct questions and analyzed (see Table 5).

### Table 9: Descriptive Statistics

<table>
<thead>
<tr>
<th></th>
<th>Total</th>
<th>Analysts</th>
<th>Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample Size</td>
<td>102</td>
<td>52</td>
<td>50</td>
</tr>
<tr>
<td>Age (mean value in years)</td>
<td>23.2</td>
<td>24.2</td>
<td>22.2</td>
</tr>
<tr>
<td>Total Work Experience</td>
<td>2.5</td>
<td>1.7</td>
<td>3.3</td>
</tr>
<tr>
<td>IT Work Experience</td>
<td>0.4</td>
<td>1</td>
<td>0.7</td>
</tr>
</tbody>
</table>

To assess construct validity we performed factor analysis. The items that did not load well were removed. Reliability analysis was then computed for each construct (see Table 6). All constructs had Cronbach alpha closer to or greater than 0.7 and thus confirmed construct validity (Cheung and Lee 2001). Process facilitation was represented by a dichotomous dummy variable (present=1/absent=0). Pearson’s product-moment correlation matrix displays significant correlations between the predictor and outcome variables (see Table 6).

A closer look at the correlation coefficients amongst Project Monitoring and Control, Communication Effectiveness and Project Success indicate partial mediation between Project Monitoring and Control and Project Success by Communication Effectiveness. Similarly, correlation coefficients amongst Project Communication, Communication Effectiveness and Project Success indicate partial mediation between Project Communication and Project Success by Communication Effectiveness. To analyze our hypotheses about the difference in perception of the outcome and the predictor variables, ANOVA was performed for all the predictor variables across the two treatments (see Table 7). The F-probability value (p<.05)
Table 10: Correlation Matrix and Reliability Analysis between the Predictor and Outcome Variables

<table>
<thead>
<tr>
<th>Items</th>
<th>Project Success</th>
<th>Project Communication</th>
<th>Project Monitoring and Control</th>
<th>Process Facilitation</th>
<th>Communication Effectiveness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Success</td>
<td>8 (.890)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project Communication</td>
<td>5 .373** (.706)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project Monitoring and Control</td>
<td>4 .688** .000</td>
<td>(.848)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process Facilitation</td>
<td>1 .266** .194</td>
<td>.050 NA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Communication Effectiveness</td>
<td>5 .684** .366**</td>
<td>.491** .207* (.839)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

** Correlation is significant at the 0.01 level (2-tailed)  
* Correlation is significant at the 0.05 level (2-tailed)  
Reliability Analysis (Cronbach alpha) indicated on the diagonal

was true for communication effectiveness with homogeneity of variance (Levene Test). Thus hypothesis H1a was supported, which implied that presence of a coordinator resulted in greater communication effectiveness but not greater project success (H1b).

Table 11: ANOVA Contrasts

<table>
<thead>
<tr>
<th></th>
<th>Levene Test (p&gt;.05)</th>
<th>F-value (df)</th>
<th>Sig. (p&lt;.05)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Success</td>
<td>.007</td>
<td>7.606 (1,100)</td>
<td>.007</td>
</tr>
<tr>
<td>Communication Effectiveness</td>
<td>.355**</td>
<td>4.461 (1,100)</td>
<td>.037**</td>
</tr>
<tr>
<td>Project Communication</td>
<td>.168**</td>
<td>3.903 (1,100)</td>
<td>.051</td>
</tr>
<tr>
<td>Project Monitoring &amp; Control</td>
<td>.841**</td>
<td>0.248 (1,100)</td>
<td>.619</td>
</tr>
</tbody>
</table>

Table 12: Results Summary

<table>
<thead>
<tr>
<th>Hypothesis Result</th>
<th>Standardized beta</th>
<th>t-value</th>
<th>Sig.</th>
<th>R²</th>
<th>Tol.</th>
<th>VIF</th>
</tr>
</thead>
<tbody>
<tr>
<td>H2a: +project monitoring and control → +perceived project success</td>
<td>Supported</td>
<td>.688</td>
<td>9.4</td>
<td>.000</td>
<td>.474</td>
<td>1 1</td>
</tr>
<tr>
<td>H2b: +project monitoring and control → +communication effectiveness</td>
<td>Supported</td>
<td>.491</td>
<td>5.6</td>
<td>.000</td>
<td>.241</td>
<td>1 1</td>
</tr>
<tr>
<td>H3a: +project communication → +perceived project success</td>
<td>Supported</td>
<td>.373</td>
<td>4.0</td>
<td>.000</td>
<td>.139</td>
<td>1 1</td>
</tr>
<tr>
<td>H3b: +project communication → +communication effectiveness</td>
<td>Supported</td>
<td>.366</td>
<td>3.9</td>
<td>.000</td>
<td>.134</td>
<td>1 1</td>
</tr>
<tr>
<td>H4: Path Analysis</td>
<td>Regression 1: Project Success (PS)= f([Project Monitoring &amp; Control (PMC) + Communication Effectiveness (CE) + Project Communication (PC)])</td>
<td>Supported</td>
<td>.524 (PMC)</td>
<td>7.8</td>
<td>.000</td>
<td>.683</td>
</tr>
<tr>
<td></td>
<td>Regression 2: Communication Effectiveness (CE) = f([Project Monitoring &amp; Control (PMC) + Project Communication (PC)])</td>
<td></td>
<td>.335 (CE)</td>
<td>4.6</td>
<td>.000</td>
<td>.625</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>.250 (PC)</td>
<td>3.9</td>
<td>.000</td>
<td>.807</td>
</tr>
</tbody>
</table>

To analyze our hypotheses about the difference in perception of the outcome and predictor variables regression models were created. Summary of the hypothesis results is presented in
Table 8. To check for multicollinearity tolerance values (greater than 0.1) and VIF values (less than 10) were considered (Ho 2006, pp. 258).

To test hypothesis 4 (partial mediation) path analysis was performed with SPSS following the procedure specified by Ho (2006, pp. 267-279). The path coefficients between project success and the three predictors (Project monitoring & control, project communication and communication effectiveness) were obtained by regressing the former on the later (see Table 10: H4 Regression 1). The path coefficients between communication effectiveness and the two predictors (Project monitoring & control and project communication) were obtained by regressing the former on the later (see Table 10: H4 Regression 2). Figure 2 presents the path model together with estimated regression coefficients (beta values) associated with the hypothesized paths. It can be stated that project monitoring & control and project communication have both direct and indirect influences on project success. This further confirms the partial mediation effect of Communication Effectiveness as pointed out earlier.

![Figure 2: Path Model](image)

**Conclusion**

In this paper we have described an exploratory quasi-experimental study, which investigates the performance of requirements analysis projects in an agile-rigid GSD environment. We constructed items for measuring various predictor and output variables relating to performance of such projects. The model was validated using a study conducted in an academic setting consisting of students at the Marquette University (MU), USA and Management Development Institute (MDI), India. Results of our study indicate that having a project coordinator increases communication effectiveness which in turn affects project success. Project communication planning, even if it is not very detailed in nature, has positive influence on communication effectiveness. From the results of path analysis (see Figure 2) it is clear that communication effectiveness mediates the effects of agile-rigid project management (project communication and project monitoring and control) on project success. The moderating effect is considerably greater for project communication. The use of GSD in organizations is becoming more and more commonplace as corporations seek to take advantage of the talent available in geographically dispersed locations for their multi-location operations. This study yields several interesting conclusions that can assist organizations in creating and managing their GSD projects more effectively. Creation of an agile-rigid environment can help organizations mitigate communication, coordination and control related risks inherent in GSD.

Like most experimental studies this research also has its limitations. However, Damian et al. (2000) argue that 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. 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). Though the quasi-experiment was carefully designed to address the controversial issue of using students as surrogates, the projects done were limited in scope and size compared to large scale industrial projects. However, no formal measures of complexity were used in the study. Our objective was to study the research questions on comparable small projects, the results of which may be applicable to projects of similar complexity. Further research is needed to assess the impact of these findings on large scale industrial projects.

Future Research Directions
One way of dealing with the lack of realism in academic experiments is to use multiple methods so that strengths of some compensate the 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 et al., 1988). Simulated laboratory negotiations could be complemented by field studies or validations, if the lack of realism is an issue. In our research therefore, we plan to complement the findings of our academic experiment with field validation. Internal validity of results was established through conducting experiments in controlled environment. We expect to conduct external validity through industry survey and case studies.

References
Carmel, Erran "Building your Information Systems form the other side of the world: How Infosys manages time zone differences" *MIS Quarterly Executive* (5:1), 2006, pp. 43-53.


