2008

Managing The Characteristics Of Software Product Development Programs

Yuzhu Li
University of Central Florida

MANAGEING THE CHARACTERISTICS OF SOFTWARE PRODUCT DEVELOPMENT PROGRAMS

by

YUZHU LI
B.S. Nankai University, 1999
M.S. Washing State University, 2002

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Management Information Systems
in the College of Business
at the University of Central Florida
Orlando, Florida

Summer Term
2008

Major Professor: James Jiang

ABSTRACT

Multiple related software development projects are often managed concurrently and systematically to deliver a complex software system. This approach of managing multiple interdependent projects together to achieve a common goal is called program management (Pellegrinelli, 1997). A software development program can generate the benefits that cannot be achieved by managing projects individually. The software product development program has the special characteristics such as complexity, uncertainty and interdependence (1995). A software product development program can play an active role in managing the uncertainty and interdependence in the software development process. This dissertation is designed to examine the external communication effectiveness of the program team on the interdependence between the program and the larger organizational context. In addition, this dissertation studies the inter-project coordination effectiveness on uncertainty within a program. Based upon organizational Information Processing Theory (IPT) and Resource Dependence Theory (RDT), theoretical frameworks are developed. The proposed research models are tested by surveying software product development programs across a range of industries. The results will contribute to the understanding of multiple-project communication in a program's context. The specific interactions between coordination/communication and the product development characteristics will provide a guideline for the industrial practices.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE: GENERAL INTRODUCTION

1.1 Introduction

Project management is now a well-established approach for organizations to carry new strategic initiatives and realize the proposed changes. When multiple projects are grouped together and managed concurrently, a program is created. This approach of grouping existing projects or defining new projects and for focusing all the activities required to achieve a set of major benefits is called program management (Pellegrinelli, 1997). Related projects in a program are managed in a coordinated way to achieve a common goal, or to extract benefits which would not be realized if they were managed independently. Programs can be built when a set of projects exist and synergistic benefits are expected to extract. Programs can also be established before starting any real projects and be used as a generational tool for a set of projects that are related and serve a strategic goal. Pellegrinelli (1997) observed that three primary reasons for creating a program are to coordinate distinct projects that share a common resource or skill base, to develop a completely new system, infrastructure or service and to enhance the existing functionality or service delivery. Of particular interests in this dissertation is a product development program that develops a completely new system, infrastructure or service. Specifically, this dissertation studies a program that develops a new software system or a product in which customized software plays a critical role in product integration. This type of program is called the product development program and several terms that are used interchangeably include: the program, software development program or software product development program.

Product development programs are usually used as a tool to carry an initiative outside of current organizational structure. Existing product updates and new product innovation are managed by the program. Changing product needs and external pressures from multiple stakeholders push the organization to take action. When no clear understanding of the product needs has been reached, the program has an ambiguous goal. Some initial trials will provide signals and feedback to the program management team. As the learning process goes, follow-up actions will be aligned with the changing environment and more projects will be started with clear objectives. Learning is a prerequisite to making progress. A supportive development environment is created in the program for projects both in exploratory nature and with identified objectives (Pellegrinelli, 1997).

Programs that have software systems as a final product or as a critical integration of the product are rather complex since a product development covers requirements management, release definitions, and new product launches. More challenges come with this complexity. Software products (or components) can be changed or updated relatively easily by using patches or release updates. Because of the supporting roles of information technology, software requirements are highly complex and changing frequently. Both internal and external stakeholders must be taken into account (Weerd, Brinkkemper, Nieuwenhuis, Versendaal, & Bijlsma, 2006). A program management team usually consists of a program manager (sometimes it is called product manager, and these two terms are used interchangeably in this dissertation), several project managers and sometimes some administrative staff and related stakeholders. This team analyzes the external and internal environments of the program and makes decisions for product development. Product managers play an essential role in program management and carry many responsibilities, but many times product managers do not have the authority over the development teams, resources and program performance evaluation (Weerd et al., 2006).

The constraints and complications of having to operate within a wider organizational context with limited resources and interdependent projects are essentially accommodated. The characteristics of software product development create enormous needs for information processing among various stakeholders. The program management team should draw resources from line managers and report to suitably senior managers to increase their power and influence. At the same time, the program management team also needs to make decisions on product development, resource allocation and inter-project coordination within the program. In this context, program management teams play a critical role in managing the internal and external environments and dealing with the characteristics of software product development through coordination and communication. This dissertation centers on Information Processing Theory and Resource Dependence Theory and examines the effectiveness of coordination and communication on the management of the characteristics of software product development in a program.

This dissertation fits in the research stream of program management (Hoegl & Weinkauf, 2005; Lundin & Soderholm, 1995; Lycett, Rassau, & Danson, 2004; Pellegrinelli, 1997, 2002, 2004; Thiry, 2004). The past literature has articulated the difference between a program and a project and the strategic advantages of program management. However no discussions have been done in terms of program operations. This dissertation contributes to the understanding of program management and develops rich implications for multiple-project operation in the Information Systems (IS) context.

Besides the program management literature, this dissertation fits in the research framework developed by Brown and Eisenhardt (1995) and extends the product development research by focusing on a unique type of product, software. Brown and Eisenhardt (1995) summarized the past product development literature into three research streams: rational plan, communication web, and disciplined problem solving. Each research stream focuses on a particular aspect of product development. This dissertation focuses on the communication web that concerns the effects of internal and external communication on product development performance. However, in the past this research stream of communication web had a narrow focus. Brown and Eisenhardt (1995) pointed out that the shortcomings of this research stream is neglecting other factors such as product attributes and market attractiveness. In addition, the past studies in this stream didn’t distinguish different types of products such as innovative products. The software development process is innovative to various extents because of uncertainty, interdependence and complexity. This dissertation tries to overcome the shortcomings of past studies in this research stream and examines the communication effectiveness on the software product development characteristics.

Two theoretical themes will be used in this dissertation. The first one is the Information Processing Theory (IPT) (Galbraith, 1973) which emphasizes that frequent and appropriately structured task communication (both internal and external) leads to more comprehensive and varied information flow. The second theoretical perspective is Resource Dependence Theory. RDT emphasizes that frequent political communication (typically external) leads to high performing development processes by increasing the resources available to product development. Based upon the these two theories, this dissertation attempts to understand how the program management team can actively manage the internal and external environment through communication and coordination.

The past literature has explored both the external and internal communication processes of a single product development team. This dissertation is going to focus on the complex software development program because of the increasing software system complexity. The past studies have indicated that external communication of a product team is critical to successful product development (Ancona & Caldwell, 1992). However software product development is different from traditional product development because of frequent release and diversified stakeholders. The external communication efforts of the program team should be strategically deployed for the special characteristics of software product development. Study I in this dissertation will focus on the external communication strategies of a program team.

Similarly, internal communication improves development-team performance in a cross-functional product development team. However, when multiple-project teams are used to develop a complex software system, the internal communication among the teams is increasingly challenging because of project boundaries, task differentiation and the innovative nature of software development. Study Two in this dissertation will focus on the communication issues within a software product development program. In addition, software development is a knowledge intensive process. Expertise is one of the most important scarce resources that the software product development program has to manage. The effective coordination of expertise can lead to team performance (Faraj & Sproull, 2000). Study Two also examines the effects of expertise coordination at an inter-project level.

1.2 Overall Contribution

This dissertation offers several significant contributions to program management literature, product development literature, software development literature and project management literature. First, this dissertation contributes to program management literature by providing the best practices of communication and coordination that a program management team can use in order to gain resources, improve the program's influence and achieve the program goals.

Second, this dissertation overcomes the shortcomings of the research stream of communication web in product development literature and includes other factors such as software development features. Including these factors will help the researchers and managers to understand in-depth the influential communication efforts and highlight the political and information-processing dynamics underlying the communication processes of successful software product development. This in-depth focus of the communication web complements the rational plan perspective of product development.

Third, this study improves the understanding of what a product development program team can do in managing software development characteristics. The program management team expands the product development program’s boundary by external communication. This dissertation studies the specific interactions between coordination/communication and software development characteristics. Managers will get specific tips on the extent of coordination on different levels of interdependence or uncertainty. The past literature has examined the competencies of program managers (Partington, Pellegrinelli, & Young, 2005) and responsibilities of program managers (Lycett et al., 2004). But most program managers are promoted from talented project managers. These project managers might not have the essential different skill sets required to perform at the position “program manager.” This study sheds some light on what a program manager should do in terms of external communication and in terms of managing multiple interdependent projects within a program.

Last, this dissertation contributes to the understanding of the large-scale software product development process. Software systems increasingly become complex to serve sophisticated business processes and needs. Software product development is critical because new products and releases are becoming the center of competition. The organizations that can employ the advanced technology and introduce the software to the market quickly will get more market share. The frequent software release can change market competition intensively. Thus the capability of managing software product development becomes a potential source of competitive advantage for many firms (S. L. Brown & Eisenhardt, 1995). This dissertation increases the understanding of managing the characteristics of software product development. The inherent complexity, interdependence and uncertainty create special challenges for program management teams. This dissertation proposes several managerial actions to manage the dynamic development process. The outcomes of this dissertation will attract the attention from top managers and inform them how to work with program management teams and facilitate the success of software product development.

A quantitative survey methodology was selected to empirically test the hypotheses developed in this dissertation. Data collection efforts include instrument validation, pilot study and data collection. The data analysis provides mixed results. The remainder of this dissertation consists of Study One, Study Two and General Conclusion.

1.3 Chapter 1 List of References

1. Ancona, & Caldwell, D. F. (1992). Bridging the Boundary: External Activity and Performance in Organizational Teams. *Administrative Science Quarterly, 37*(4), 634-661.
2. Brown, S. L., & Eisenhardt, K. M. (1995). Product Development: Past Research, Present Findings, and Futuer Directions. *Academy of Management Review, 20*(2), 343-378.
3. Faraj, S., & Sproull, L. (2000). Coordinating Expertise in Software Development Teams. *Management Science, 46*(12), 1554.
4. Galbraith, J. R. (1973). *Designing Complex Organizations*. Reading, Mass.: Addison-Wesley Pub. Co.
5. Hoegl, M., & Weinkauf, K. (2005). Managing Task Interdependencies in Multi-Team Projects: A Longitudinal Study. *Journal of Management Studies, 42*(6), 1287-1308.
6. Lundin, R. A., & Soderholm, A. (1995). A theory of the temporary organization. *Scandinavian Journal of Management, 11*(4), 437-455.
7. Lycett, M., Rassau, A., & Danson, J. (2004). Programme management: a critical review. *International Journal of Project Management, 22*(4), 289.
8. Partington, D., Pellegrinelli, S., & Young, M. (2005). Attributes and levels of programme management competence: an interpretive study. *International Journal of Project Management, 23*(2), 87-95.
9. Pellegrinelli, S. (1997). Programme management: organising project-based change. *International Journal of Project Management, 15*(3), 141-149.
10. Pellegrinelli, S. (2002). Shaping context: the role and challenge for programmes. *International Journal of Project Management, 20*(3), 229.
11. Pellegrinelli, S. (2004). The difference between project and programme management. *Computer Weekly*, 36.
12. Thiry, M. (2004). "For DAD": a programme management life-cycle process. *International Journal of Project Management, 22*(3), 245.
13. Weerd, I. v. d., Brinkkemper, S., Nieuwenhuis, R., Versendaal, J., & Bijlsma, L. (2006). *Towards a reference framework for software product management* Paper presented at the 14th IEEE International Requirements Engineering Conference (RE'06).

CHAPTER TWO:
WORKING IN AN INTERDEPENDENT PRODUCT DEVELOPMENT PROGRAM: THE ROLE OF EXTERNAL COMMUNICATION

2.1 Introduction

The importance of agents in product development has been explored in the literature. Through a thorough literature review, Brown and Esisenhardt (1995) claimed that process efficiency and product effectiveness are affected by the behavior of different agents, including team members, project leaders, senior managers, customers and suppliers. They consider agents primarily responsible for performance improvements. But they argue that multiple players influence product performance (S. L. Brown & Eisenhardt, 1995). Verona (1999) further explored the role of agents and argues that while acknowledging the direct contribution of agents to product development performance, one must also observe that part of their actions focuses on leveraging organizational capabilities. Verona (1999) separates the contribution of agents and capabilities from the final outcome of the product development process and concludes that the final performance of product development can be driven by both the presence of peculiar agents and their leveraging of organizational capabilities.

Software product development has inherent interdependence with the functional lines in an organization. This interdependence inhibits the leveraging of organizational capabilities. The product development success depends on how the agents overcome the interdependence in software product process and leverage the organizational capabilities. However the role of agents who form the organizational capabilities through activities and decision-making has been under-studied (Verona, 1999). This study tries to fill in this research area and examines how the agents leverage organizational capabilities by performing activities and making decisions to drive the outcomes of product development.

Agents in the product development process can refer to product designer, product manager, project manager and team leader, etc. When multiple interdependent projects are organized together, a program management team is deployed to carry the program management responsibilities such as resource optimization, product integration and coordination (Pellegrinelli, 2002). The management team plays the role of an agent and exercises more influence than any individual agents. The program manager clarifies ambiguous goals for the program, and aligns the program’s goals with the organizational strategic initiatives. This person also carries the responsibilities of communicating the program status with the top managers, lobbying resources for the program by persuading the functional department heads and coordinating multiple project schedules in a centralized way. The project managers in this team are coordinating the product development issues that affect the project inputs and outputs within the program, share knowledge and seek solutions for the project issues that arise because of special characteristics of software product development.

A program management team is a more powerful agent in the organization than an individual product manager or project manager or even any senior manager. Programs set up a context for individual projects by grouping them, directing them and initiating them. Programs set the boundaries within which projects managers can operate, protecting them to an extent from external pressures and uncertainties. Programs build themes, identify intentions of external influence, and translate those intentions into concrete objectives for projects. Moreover, programs have a stronger representation than individual projects in an organizational structure. A program can obtain more resources, have larger bargaining power and increase the visibility to top management. Sufficient support from stakeholders can smooth the strategic implementation process and lead to program and project success. In addition, program management can also enhance the quality of decision making because it considers multiple perspectives and makes decisions based on changing business needs and environment uncertainty.

Since the program management team has to clarify the ambiguous program goals, lobby for resources and manage multiple stakeholders, the external communication strategies proposed by Ancona and Caldwell (1992) fit in the context very well. Ancona and Caldwell (1992) classify the external communication strategies of a product development team into four types: ambassadorial strategy, task coordinator strategy, scouting strategy and guarding strategy. To further the understanding of the program management team, three activities of the ambassadorial strategy - filtering, molding and mapping - are particularly examined in this study.

Filtering has a protective goal and it refers to the activities that are conducted by the program manager and will filter the pressure and requests from the organizational environment and isolate the product development from the excessive pressure so that the product can be developed without too many barriers. Molding refers to activities with the purpose of informing the functional lines in the organization about the product development status and convincing others that the product development is important for other business functions and can benefit the whole organization. Mapping refers to activities with the purpose of involving the senior managers in making critical decisions and forcing the business functions to support the product development because of the hierarchical orders.

An organization builds the product development program and provides the resources but sets up constraints at the same time. This paper argues that the team leverages the organizational capabilities embedded in the structures and systems and overcomes the constraints by influencing the relationship of the product development program and the larger organizational context through external communication activities.

This paper picks up several important organizational capabilities such as goal interdependence, reward interdependence (e.g. incentive systems), and resource interdependence (see Table 1 for definitions) and looks at how the agents can leverage the organizational capabilities through external communication. These interdependencies are important for the program management team because they are related to the motivation of other functional departments to communicate and support the product development program.

The specific research question is "How does the product development program communicate with the external environment to deal with the different types of interdependence between the program and the organizational internal environment?" This paper proposes the best communication strategies that a product development program should utilize to take advantage of the organizational capabilities and get around the organizational constraints so that the product can be successfully delivered at the end.

Literature review on external communication, interdependence and agents' role are done in Section 2.2. A theoretical framework is proposed in Section 2.3 and best communication strategies are proposed for each type of interdependence. Hypotheses are developed in Section 2.4. Research methodology including constructs and data collection is reported in Section 2.5. The data analysis results are presented in Section 2.6. Discussions and conclusion are developed in Section 2.7 and 2.8.

2.2 Literature Review

2.2.1 Interdependence

The failure rate of software development projects has been high. According to the Standish Group study, the success rate of software development project was only 16% in 1995 (Standish, 1995). It has been improving and is estimated to be 34% in 2002 due in part to project management since 1995 (Standish, 2002). However the Standish Group study in 2006 reports that 35% of software projects can be categorized as successful meaning that they were completed on time, on budget and met user requirements (Rubinstein, 2007). The success rate was improved but not to a large extent because of the changing business needs and increasing complexity of software systems. Kraut and Streeter (1995) have explored the software development characteristics and identified three major features: uncertainty, complexity and interdependence.

Uncertainty, complexity and interdependence are the sources of coordination (Galbraith, 1973). Large size and uncertainty in software development would be less of a problem if software didn't require precise integration of its components. Poor coordination between subgroups producing software modules could lead to failure in integrating the modules themselves. The major studies of interdependence in IS literature have focused on either task interdependence in the software development team (Andres & Zmud, 2001) or interdependence between the software developers and users such as the user involvement literature. However when software development process is viewed as a product development, more types of interdependencies should be studied in this particular context.

The interdependence of the product development with the other organizational functions is of particular interest in this paper. Product development is embedded in a large reciprocity-based intra-organizational relationship. Due to imbalances in the distribution of resources, attention and authority, and because of the interdependencies among projects and organizations, the environment in which product development exists is described as uncertain (Jensen, Johansson, & Lofstrom, 2006). As a consequence, product development program managers must use different kinds of strategies to attract attention, enroll stakeholders, and to mobilize support from more distant but powerful actors.

Table 1 lists different types of interdependencies. Resource interdependence is task-specific. The goal and reward interdependencies belong to the incentives and rewards systems that are decided by the top managers. The incentive and rewards systems will positively impact the internal integration. Although interdependence from the organizational structure is predominant in the product development environment, it changes as the external market and the organizational business strategy change.

Table 1 Definitions of three types of interdependence

Construct	**Definition**	**Studies**
Goal Interdependence	the degree to which programs have clear goals or a clearly defined mission, and the extent to which the goals of the program are linked to other organizational units.	(Ortiz, Johnson, & Johnson, 1996); (Campion, Medsker, & Higgs, 1993; Campion, Papper, & Medsker, 1996)
Resource Interdependence	the extent to which one organizational unit need certain resources that are only available from other unit.	(Gattiker & Goodhue, 2005a; Kim, Umanath, & Kim, 2005; Tushman & Scanlan, 1981; Wageman, 1995) (Sharma & Yetton, 2003)
Reward Interdependence	the degree to which shared significant consequences of product development are contingent on collective performance of the program and the related functional lines and programs.	(Fan & Gruenfeld, 1998; Wageman, 1995)

Interdependence has been intensively studied in the group/team literature. In research with traditional teams, all three aspects of interdependence have been shown to be positively related to motivation of team members and team effectiveness (Campion et al., 1993; Campion et al., 1996; Locke & Latham, 1990; Wageman, 1995).

Goal interdependence is defined as the degree to which programs have clear goals or a clearly defined mission, and the extent to which the goals of the program are linked to other organizational units (Campion et al., 1993; Campion et al., 1996). The productivity and achievement of a group are largely dependent upon the degree of positive interdependence that exists among the group's members (Johnson, Johnson, & Stanne, 1989). At the group level, goal interdependence links the individual success to group success. Goal interdependence motivates individuals to help each other in the interest of group productivity, because they, as individuals, will benefit. At the organizational level, when a product development program and other business functions have a high level of goal interdependence, these business units are motivated to provide support and feedback to the product development so that products can be delivered in a short time and have a good quality.

In contrast, *Resource interdependence* is the extent to which group members control or influence one another's access to critical materials or resources (Pfeffer & Salancik, 1978), such as data, physical materials, unique or exclusive information and knowledge, and/or specialized skills, abilities, and performance behaviors. Resource interdependence motivates individuals to elicit and use others' resources to achieve personal goals (Johnson et al., 1989). Groups with "high" resource interdependence must work together in order to accomplish their task, whereas groups with "low" resource interdependence can accomplish their collective goals by working independently. In this case, individuals benefit results from the acquisition of others' resources, not from their success. Therefore goal and resource interdependencies may not elicit equivalent levels of cooperation and may even affect productivity differently (Johnson et al., 1989). At the organizational level, inter-unit resource flow is more difficult than the inter-personal resource acquisition. Each unit has only limited resources and multiple goals. When the resource interdependence between the product development program and the business functions such as marketing is high, the program team has to spend the time and effort to coordinate and communicate so that the sufficient resources can be obtained on time. Sometimes resources from other business units are acquired through formal work processes and at some occasions the resources are delivered through informal exchange. The program team has to make use of all the possible means to manage resource acquisitions from the external sources.

Reward interdependence is the extent to which team members' outcomes (e.g., praise, bonuses, organizational recognition, compensation) are contingent on the performance of the group as a whole rather than their individual achievements (Wageman, 1995). The "higher" the level of reward interdependence, the more that rewards are based on group (as opposed to individual) performance. At the organizational level, the high level of reward interdependence can motivate the other business units to transfer information and resources to the product development program because the rewards are based upon the overall organizational outcomes such as sales and profits instead of the solo individual department’s performance. In an organization with a high level of reward interdependence across the product development program and other organizational units, the product development program’s rewards depend not only upon the outcomes of the product development program but also upon the ability to coordinate and integrate the performance and information into the product development process so that the overall organizational benefit can be achieved.

2.2.2 The Role of the Program Management Team

A software product development program is usually considered one level up along the management ladder in organizations than any single project. It is intermediary between the strategic level and the operational level. Multiple projects are implemented for achieving one strategy initiative. Programs have been developed for single-minded focuses. Projects focus on the definition, planning and execution of specific objectives.

Programs set up a context for individual project operation by grouping them, directing them and initiating them. Programs set the boundaries within which project managers can operate, protecting them to an extent from external pressures and uncertainties. Programs build themes and identify intentions of external influence, and translate them into concrete objectives for projects. Moreover, programs have a stronger representation than projects in organizational structure. Program can obtain more resources, have larger bargaining power and increase the visibility to top management. Sufficient support from stakeholders can smooth the strategic implementation process and lead to program and project success. In addition, program management can also enhance the quality of decision making because it considers multiple perspectives and makes decisions based on changing business needs.

Program managers play a critical role in representing the product development program in the organization, conduct external communication, and manage the internal coordination between multiple projects within a program. The overall responsibility of program managers is to realize the anticipated benefits from the program. The major work contents of program managers are to set objectives for projects, facilitate interactions among project managers, monitor project progress, collect reports from project managers, organize a comprehensive report and present to the senior management. The unit of implementation within a program is a project. While project managers manage the process of delivering the specified project outcomes, they participates in the program management activities including monitoring other projects' progress, allocating resources, identifying uncertainty and risks, and appraising project and program performance.

The formal position of program manager along the hierarchical line gives the program manager opportunities to talk to the senior managers and integrate organizational competencies. Program managers have the capability to mediate between the program and the large organizational context. Program managers administer the program’s operational tasks. This mediating role enables them to have the potential to influence perceptions of the senior managers and to change their strategic priority (Floyd & Wooldridge, 1992).

Their potential to influence comes from not only the formal position and but also their competencies. The program manager’s competencies have to range from “focus on details only” to “appreciation of contextual and future consequences” along four levels (Partington et al., 2005) from detail operations to overall pictures. Program managers have to manage the three relationships including *self and the work*, *self and the others*, and *self and program environment* (Partington et al., 2005). Program managers use social and parental influence to coach the program team and inspire the team to learn and adapt to new environmental changes. In dealing with the relationship “self and program environment”, program managers are aware of shortcomings of current operations and prepared for contingencies. Program managers provide analysis and opinions in consistent style and sell vision of outcome by remaining sensitive to the audience. Therefore program managers can influence the strategic agenda in the organization. By performing these influential activities, program managers can provide a supporting environment for the program development and make program performance success more likely.

2.2.3 Communication

According to IPT (Galbraith, 1973), the program team must exercise the extent of communication and coordination based upon the needs created by interdependence with functional lines in the organization. This external communication spans the boundary of the program. Organizational boundary spanning literature argues that organizational decision makers must have the information about environmental contingencies to make appropriate decisions relevant to the environmental conditions and contingencies (Leifer & Delbecq, 1978). The impetus for boundary spanning activity (BSA) comes from any number of sources, such as multiple goal structure, nonroutine technology, perceived environment complexity and/or instability. These require BSA to reduce the uncertainty and inability to make decisions based on available information which generates information search etc. BSA is viewed as the intervention between environmental characteristics and organizational processes and functioning. Thus when studying boundary spanning, one must be more sensitive to the local contingencies on those boundary spanning persons and processes than to some larger, overall organizational characteristics which may have little predictive or explanatory power (Leifer & Delbecq, 1978).

Although product development is affected by the overall organization and industry, only the local organizational environment and the particular product market can affect the product development program. Product development teams attempt to influence the larger organization by managing their interfaces with the larger organization through various activities and strategies (Ancona & Caldwell, 1992). The product managers have to manage the local environment successfully to get resources and supports for product development.

The product development team’s externally focused communication can be categorized into four major types of strategies (Ancona & Caldwell, 1992). “Ambassador” strategy involves frequent communication with managers above the team in the organizational hierarchy because the team lobbies for resources and seeks protection and support. "Task coordinator" strategy is carried out to coordinate technical or design issues and is often conducted laterally across the organization. "Scouting" strategy is conducted to scan for ideas regarding the competition, technology, or the market in general, and is aimed at specific functions within the firm such as sales and marketing. The fourth type of strategy is "guarding" strategy which is performed to prevent the release of information to external entities. Since the guarding activity is an internal activity with external focus, it will be excluded from this study.

A product development program team should perform the communications with the similar strategies to manage the external environment. Three activities of ambassadorial strategy including filtering, molding and mapping are considered here. The software product development program manager will play a critical role here.

2.3 Theoretical Framework

Product development is a process of integrating different functions and required knowledge to deliver a product that satisfies the customers’ needs. Product development projects cannot be isolated from the organizational environment since projects depend on different kinds of resources such as money, time, knowledge, reputation and trust, etc. Since no project is completely self-contained, the key to survival is the ability to acquire resources (Jensen et al., 2006). The environmental settings of the product development projects are often described as highly political; the diversity of interest and competition gives rise to “wheeling and dealing”, negotiation and other processes of coalition building (Platje, Seidel, & Wadman, 1994). These multi-project environments are characterized by a high degree of uncertainty since the projects have to compete for scarce resources.

A product development program consisting of multiple interdependent projects builds a protective layer for projects and represents the related projects as a bigger and more important entity in the organization than individual projects. The sources for the product development program’s uncertainty are usually business changes. The product development program gets affected because of the interdependence with the business departments in the organization. The product development program’s management team usually translates the business changes into specific goals and system requirements for the product development team.

The interdependencies between the product development program and internal organizational environment are represented by goal interdependence, resource interdependence and rewards interdependence. These interdependencies force external communication to become one of program management team’s major focuses. The goals of external communication are the acquisition of resources and supports from functional lines and senior managers. Resource Dependence Theory (RDT) proposes that agents lacing in essential resources will seek to establish relationships with others in order to obtain needed resources (Tillquist, King, & Woo, 2002; Ulrich & Barney, 1984). Originally, RDT was formulated to discuss relationships between organizations. However the theory is applicable to relationships among units within an organization. The program management team tries to alter their dependence relationships by minimizing their own dependence or by increasing the dependence of others on them. Within this perspective, the program management team is viewed as coalitions altering their structure and patterns of behavior to acquire and maintain needed external resources. Acquiring the external resources needed by the program comes by decreasing the program’s dependence on others and/or by increasing other’s dependency on it. In other words, the program team is modifying a program’s power with the functional lines in the organization.

The modification of the dependence with other function lines is completed by external communication. Reward interdependence can be changed by convincing the senior managers and setting up rewards for supports of the program. Goal interdependence can be shaped and molded by persuading the department heads from the functional lines about the importance of the program and the benefits that the functional lines can get from the program’s success. The impact of resource interdependence on the program development can be reduced if a close relationship between the program and the functional lines exists. Task coordination activities will work out the resource dependence issues as well. Ancona and Caldwell (1992) argued that these external activities enable the team to acquire resources, influence their stakeholders and lead to project success. As an organizational entity which has bigger presentation and more power than traditional individual teams, product development program should also be able to perform these four types of activities that can lead to successful product development.

These external communication activities have to be done to acquire the resources and supports for the program. However, the program team has limited resources for external communication and only certain efforts can be spent on external communication. According to the Information Processing Theory (IPT) (Galbraith, 1973), the extent of external communication activities depends on the needs for information processing such as interdependence. When multiple goals exist in the program and organization, a high level of external communication performed by the product development program team will clarify the priority of goals that the program must achieve. The team will also clarify the measures by which that program is evaluated. When a large amount of resources needed for product development are located outside of the program, a high level of external communication is necessary. This communication is performed by the program management team who will lobby for the resources. The extent of external communication will have a moderating effect on the relationship between interdependence and the program performance. A theoretical framework is proposed in Figure 1.

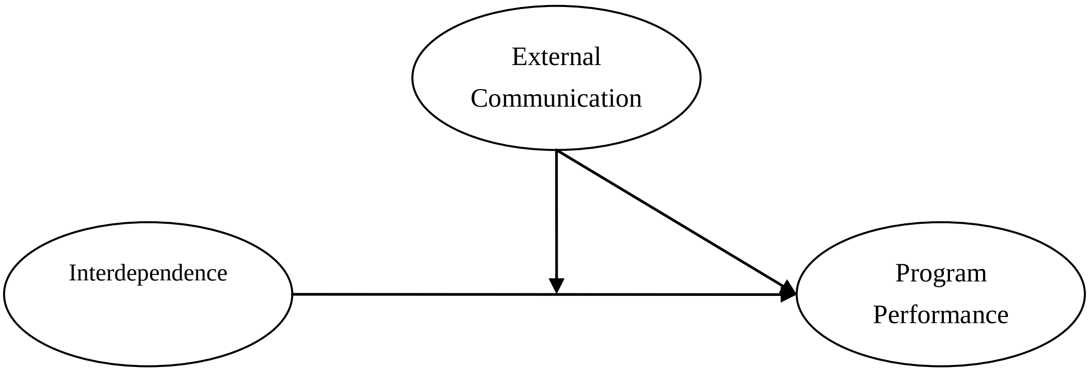


Figure 1: Theoretical framework of the program’s external communication

A program management team has access to the power structure of the organization as members of program promotion, the securing of resources, and team protection from excessive interference. The team has access to the workflow structure, which assists with managing horizontal dependence. Through coordination, negotiation, and feedback, the program is more tightly coupled with other organizational units. Mapping activity provides access to the information structure; it is aimed at adding to the expertise of the group. These activities allow the group to update its information base, providing new ideas about technologies and markets. Although these communication activities can lead to positive product development outcomes, communication also creates a high level of costs in the product development processes. Communication efforts must be deployed strategically so that the effects of communication can be maximized to deal with interdependence and uncertainty. Focusing on each type of interdependence and uncertainty, this paper proposes a set of best communication strategies for the product development program (see Table2). These communication strategies can be used to address the need of communication from the organizational structure and produce an efficient and effective product development process.

Table 2: The proposed best communication strategies

Contextual Factors	**External Communication Activities**
Resource Interdependence	Molding activity, mapping activity, Task Coordinator Activities
Goal Interdependence	Filtering activity, Molding activity, mapping activity
Rewards Interdependence	Filtering activity, molding activity, mapping activity

2.4 Hypotheses Development

2.4.1 Goal Interdependence

A product development program attracts more attention from top management than common project teams. But a program still has to compete with other programs and functional departments for limited resources. The strategy literature points out organizational strategies are adapted to changing external and internal environments. There is always some gap between the strategic level and the operational level. The program's goals are often not clear, and they are often adjusted as the business' needs change.

According to the small group theory, goal interdependence can enhance the collaboration between team members and positively impact the group's performance (Locke & Latham, 1990). At the organizational level, goal interdependence specifically refers to the extent to which the goals of the product development program are clear and the extent to which the program's goals are linked with the goals of other business functions in the organization. This goal interdependence between the program and the organizational context leads the program to actively seek clarification of its goals, reduce ambiguity, and attempt to influence the external environment to suit its agenda by shaping the beliefs and behaviors of outsiders.

When goal interdependence is high, there will be more motivation to cooperate and exchange resources and information (Vancouver & Schmitt, 1991) which can contribute to the final success of product development. However, when goal interdependence is low, other business functions in the organization have a low level of motivations to provide resources or feedback to the program and its product. The lack of resources and supports will be detrimental for the success of product development. The program management team's ambassadorial activity can develop more communication with other business functions, clarify the ambiguous program goals, align the program goals with the organizational and departmental goals, build coalitions, and persuade others about the importance of the program. In essence, these persuading and influencing activities are part of the program’s capability to implement the organizational strategy and fill in the gap between the strategic and the operational level.

Molding and mapping are the appropriate activities that program managers should perform to manage goal interdependence. When goal interdependence is high, a high level of molding activity will shape the beliefs of the functional lines and put the program in a positive image. The program is believed to make important contributions to both the functional departments and the organization. The clarified program goals are now aligned with the organizational and the departmental goals. When goal interdependence is low, the beliefs of other functional lines have less influence on the program. Less molding activity in the conditions of a low level of goal interdependence will save the program management team’s efforts. This will enable them to allocate time and attention to other important program issues so that the program performance can be high. Therefore, the moderating effect of molding activity on goal interdependence is hypothesized in Hypothesis 1.

> *H1: The magnitude of the relationship between goal interdependence and program performance is moderated by the extent of molding activity.*

Mapping is an important external communication activity that the program management team must perform to construct an overall picture of the external environment including predicting future trouble spots or potential allies for the program. Mapping is often done by combining the information from team members’ prior experiences with the information gleaned from the conversation with outsiders. A personal network of the team members in the program management team plays a critical role in mapping and collecting information. When goal interdependence is high, the functional lines have a high level of motivation to provide resources and support product development. Therefore a high level of mapping will be a waste of the program's resources and generate small or even negative impacts on the program performance than a low level of mapping. When goal interdependence is low, a low level of mapping will be sufficient for the program team to keep an eye on the external environment and remain aware of the environmental changes. Consequently H2 is proposed as follows:

> *H2: The magnitude of the relationship between goal interdependence and program performance is moderated by the extent of mapping activity.*

A high level of goal interdependence motivates the functional lines to watch product development closely. Achieving good product quality benefits both the program and the functional departments. When goal interdependence is low, the requests from functional lines that do not share common goals might conflict with the product plan and create burdens for product development. The program management team communicates with the related functional frequently and receives a large of amount of information. However, if the program team passes all the information and requests to the product development teams, the teams will be confused and cannot make decisions. The program performance will be seriously affected because of conflicting requests and resource waste on the large amount of information processing. When goal interdependence is high and the program management team just needs to make sure the information consistency and pass to the product development teams.

However, when goal interdependence is low, the functional lines are not motivated to pay attention to product development. The program management team should filter the information and only passes the appropriate information to the project teams so that the projects can concentrate on project execution and accommodate the changes in an effective way. Therefore it is proposed that

> *H3: The magnitude of the relationship between goal interdependence and program performance is moderated by the extent of filtering activity.*

2.4.2 Reward Interdependence

Reward interdependence refers to the degree to which shared significant consequences of product development are contingent on the collective performance of the program and the related functional departments and programs (Wageman, 1995). Reward interdependence includes both tangible outcomes that accrue to the group as a whole and intangible outcome such as reputation. Gladstein (1984) found that rewards, in the form of pay and recognition, had their largest influence on how the group leader behaved and how the group set itself up to work. Theoretically, high reward interdependence encourages team members to cooperate and reduce the incentives for competition.

There are imbalances in the distribution of resources, attention, and authority in the organization. When reward interdependence is low, it is difficult for the product development program to get the resources and support from other business departments to help the product development. A high level of ambassadorial activities can attract attention, enroll stakeholders, and mobilize support from more distant but powerful actors. Trust can be an important factor that shapes and stabilizes the relationship between the product development program and other business units (Wageman, 1995).

When reward interdependence is low, a high level of molding activity should be performed to change the beliefs of related functional lines and motivate the departments to provide the resources to the program. A high level of molding activity also includes talking to the senior managers, shaping their perceptions of the program and pushing them to reward supporters of functional lines. When reward interdependence is high, a low level of molding is needed for program performance. Therefore it is proposed that

> *H4: The magnitude of the relationship between reward interdependence and program performance is moderated by the extent of molding activity.*

When reward interdependence is low, a high level of mapping will enable the program management team to construct a picture of the external environment and find out possible allies. The program manager can also build coalitions through political powers that can increase the supports for the product development. The program manager can get resources by personal ties with other business units when a clear map of the external environment is generated. Therefore, when reward interdependence is low and when a high level of mapping activity is performed, other business departments will have more motivation to cooperate with the program and have less competition for the resources required for product development. The following hypothesis is proposed:

> *H5: The magnitude of the relationship between reward interdependence and program performance is moderated by the extent of mapping activity.*

Filtering activity consists of taking information from outsiders and delivering a small amount to the group. Often filtering is done to buffer the program and to absorb the external pressure from related stakeholders by too much information or political maneuvering. When reward interdependence is high, the related stakeholders and functional lines keep a high level of interests in product development and intervene to maximize the benefits out of the product development process. Filtering will enable the program to build a protective layer for the individual projects and assure the product to be delivered on time. Therefore when reward interdependence is high and the level of filtering activity is high, the program can be executed in accordance with the plan with certain flexibility. When reward interdependence is low, the low level of filtering will save the program team's resources to solve other issues. Therefore it is proposed that

> *H6: The magnitude of the relationship between reward interdependence and program performance is moderated by the extent of filtering activity.*

2.4.3 Resource Interdependence

Acquiring resources is the key to survival for product development. Deployment of resources in initial stages of product development is one of the important drivers for the product quality (Krishnan, Kriebel, Kekre, & Mukhopadhyay, 2000). In organizations, resource interdependence varies depending on the purpose for which groups are composed. The product development program pools differing areas of expertise to accomplish the goal of deliver a product successfully. However the program does not have the exclusive resources that the product development needs. In addition, because the program adapts to the strategic changes, the needs for resource from the organizational environment are changing.

When resource interdependence is low, the product development program can accomplish the goals without access to other business functions' resources. When resource interdependence is high, asking and negotiating for needed resources performed by the program management team will lead to necessary resource acquisition for product development. In addition, when resource interdependence is high, other related stakeholders have legitimate concerns regarding the product development and will try to push the product development towards the "right" directions in their minds. Multiple views are included in the decision making process along with the product development process (Fan & Gruenfeld, 1998). The high level of ambassadorial activities will keep the stakeholders updated about the product development and will set up appropriate expectations for the product that the program will deliver. Therefore the following hypothesis is proposed.

> *H7: The magnitude of the relationship between resource interdependence and program performance is moderated by the extent of molding activity.*

A program management team has to scan the internal organizational context and identify the allies who can support the program in terms of resources, influences, support and program performance valuation. When resource interdependence is low, a high level of mapping activity is unnecessary. When resource interdependence is high, a high level of mapping activity will enable the team to build coalitions and lobby the resources for product development. Therefore it is proposed that

H8: The magnitude of the relationship between resource interdependence and program performance is moderated by the extent of mapping activity.

The project managers in the product development program perform the task coordinator activity aimed at coordinating technical or design issues such as discussing design problems with others, obtaining feedback on the design, and coordinating and negotiating with outsiders. This task coordinator activity is conducted horizontally to deal with the horizontal relations in operational work processes in order to perform an assigned task. That is, relations that do not include supervision, control, or evaluation (Jensen et al., 2006).

When the level of resource interdependence is low, a high level of task coordination is not necessary since the product development teams can make decisions by themselves. Furthermore, coordination also has a cost. When resource interdependence is high, a high level of the task coordination activity between the product development program and other business functions will lead to good product quality. Coordinating usually involves resolving the issues of interdependent schedules and product designs. Although the focus of this activity may be integrating work schedules and product designs, there is often negotiating going on as well. This negotiating is particularly common because of shifting power and dependency relationships between programs and other parties. Based upon the previous argument H9 is proposed.

H9: The magnitude of the relationship between the resource interdependence and program performance is positively affected by the extent of task coordinator coordination activity.

2.5 Research Methodology

A survey was used to collect data and, from it, the hypotheses were tested.

2.5.1 Measures

All the items were measured on a five-point Likert scale, ranging from "to a large extent" (5) to "not at all" (1). All the survey items are listed in Table 11 in 2.9 Appendix. The constructs were:

a. Program external activities: Twenty-four items of the program teams' external activities were adapted from Ancona and Caldwell (1992). The ambassadorial strategy has twelve items. The program management team performs this set of activities to protect the program from outside pressure, to persuade others to support the team and to lobby for resources. Task coordinator activity has five times. Examples include discussing design problems with outsiders, obtaining feedback on the product design, and coordinating and negotiating with outsiders.

The ambassadorial strategy includes several sub-activities (Ancona & Caldwell, 1992). Examining the twelve items for the ambassadorial activity, I further divided the twelve items into three categories. Each category reflects different communication purposes including molding, filtering and mapping and therefore each activity measure is reflective. The filtering activity has three items. The molding activity has 4 items and the mapping activity has four items. The measure of ambassadorial strategy is formative because the individual items describe and define the construct instead of reflecting the construct (Petter, Straub, & Rai, 2007). Although filtering, molding and mapping activities are reflective, they form the construct of the ambassadorial strategy. Allll the VIF statistics between the sub-activity and the construct are less than .30 (Petter et al., 2007).

b. *Interdependence*: The measure for resource interdependence has six items from Brown et al. (M. M. Brown, J. O'Toole, & Brudney, 1998) and two of them were adapted to suit the program context. The measure of goal interdependence has three items adapted from Pearce and Gregersen (1991). The measure of reward interdependence has three items adapted from Gattiker and Goodhue (2005b).

c. *Program Performance*: Program performance is measured by product flexibility which is one of the important dimensions of product quality (S. Nidumolu, 1995). When a product development program is deployed to execute the product development process, product flexibility can be achieved to a larger extent than the product developed in an individual project. The product flexibility construct has three items adapted from Nidumolu (1995). An example item is “overall long term flexibility of the product.” The specific items are provided in Appendix. Each item was included in the questionnaire and scored using a five-point scale ranging from “disagree” (1) to “agree” (5).

Table 3: List of Construct in Study I

Construct	**Dimensions**	**Loadings**	**T-stat**	**Item-Construct**	**Cronbach Alpha**	**Composite Reliability**	**Variance Extracted**
Goal Interdependence	god1	0.96	84.46	0.84	0.91	0.96	0.92
	god2	0.96	84.46	0.84			
Resource Interdependence	rsd3	0.82	15.98	0.58	0.78	0.89	0.73
	rsd4	0.86	15.06	0.60			
	rsd6	0.87	24.14	0.69			
Reward Interdependence	rwd1	0.86	19.23	1.00	0.63	0.85	0.74
	rwd2	0.86	19.23	0.60			
Filtering	Compur1	0.92	50.75	0.70	0.83	0.92	0.85
	compur2	0.92	50.75	0.70			
Molding	compur5	0.72	6.65	0.57	0.79	0.86	0.6
	compur6	0.82	20.49	0.64			
	compur11	0.70	4.60	0.50			
	compur12	0.86	28.70	0.75			
Mapping	compur10	0.80	14.09	0.60	0.77	0.86	0.61
	compur4	0.72	7.00	0.54			
	compur8	0.81	12.28	0.59			
	compur9	0.77	7.24	0.60			
Task Coordination	compur13	0.87	34.13	0.67	0.73	0.85	0.66
	compur14	0.76	11.34	0.50			
	compur17	0.80	7.91	0.54			
Product Flexibility	pp16	0.77	10.57	0.52	0.84	0.81	0.58
	pp17	0.74	8.12	0.78			
	pp18	0.77	10.10	0.79			

2.5.2 Data Collection

Data was collected in China in 2007-2008. The data collection unit is a “program”. On average, each program includes 3-5 individual IT projects. For each program, a program manager was identified and invited to fill in the questions about the program team’s external communication and interdependence. A project manager who worked with the participating program manager answered the questions about product flexibility. The recruiting method for participants was snowballing. The investigators' friends who worked in a product development program were invited to participate in the survey. They were requested to introduce more participants. Fifty-six pairs of program managers and project managers completed surveys and fifty-three pairs are valid. This small sample size causes the issue of statistical power.

2.6 Data Analysis

2.6.1 Demographic Information

The demographic information of respondents is shown in Table 4. Of those participants who provided gender information, 90.6% were male and 7.5% were female. The respondents consisted of 28 program managers, 6 project managers and other managers who oversee the development of a software product development. The average year of experience in the IT industry was 9.12 years and the average year of experience in the current company was 5.04 years. Each respondent had 22 subordinates on average. Among the participants, 47.2% came from IT industry. About 42.9% of the companies had employees less than 500 but more than 50. About 18.6% of the companies had more than 1000 employees. 30.2% of the programs were completed less than 1 year and 50.9% were completed in less than 2 years. Among the respondents, 24.5% received PMP certification and 34.0% were currently pursuing the certification. Overall, the pool of respondents was well qualified to judge the issues related to external communications and product development performance.

Table 4: Demographic Data of Program Managers in Study I

Variables	Categories	Number	Percentage
Gender	Male	48	90.6
	Female	4	7.5
	Missing	1	1.9
Position	Project Leader	2	2.9
	Project Manager	6	15.7
	Program Manager	28	47.1
	Product Director	2	12.9
	Product Manager	3	0
	IT Director	2	1.4
	VP	3	8.6
	Others	1	8.6
	Missing	6	11.3
Industry	IT industry	25	47.2
	Non-IT industry	28	52.8
Company size	<=50	11	27.1
	50-500	20	42.9
	500-1000	9	8.6
	>1000	11	18.6
	Missing	2	3.8
Average project duration	< 1 year	16	30.2
	1-2 years	27	50.9
	2-3 years	4	7.5
	3-5 years	3	5.7
	>=6 years	1	1.9
	Missing	2	3.8
PMP certification	Certified	13	24.5
	Pursuing	18	34
	Intend to pursue	3	5.7
	Not certified	14	26.4
	Missing	5	9.4
Work experience		9.12yr	
Current company experience		5.04	
Number of subordinates		22	
Total Sample size		53	

2.6.2 PLS analysis

The hypotheses were tested and verified by using the partial least square (PLS) analysis (Lohmoller, 1989). This is a latent structural equation modeling technique that uses a component-based approach to estimation; it contains two steps. The first examines the measurement model and the second assesses the structural model. When using PLS, researchers must pay attention to three concerns: (1) the reliability and validity of measures, (2) the appropriate nature of the relationship between measures and constructs, and (3) the path coefficient, model adequacy, and the final model from the available set of alternatives (Hulland, 1999). The PLS-Graph Version 3.00 was used to test the hypotheses.

2.6.3 Measurement Model

Item reliability, convergent validity, and discriminant validity tests are often used to test the measurement model in PLS. Individual item reliability can be examined by observing the factor loading of each item. High loading implies that the shared variance between the construct and its measurement is higher than the error variance. Factor loading higher than 0.7 can be viewed as highly reliable and factor loadings less than 0.5 should be dropped. Table 3 has shown the factor loading, item-total relationship and T-statistics.

Convergent validity is assured when multiple indicators are used to measure one construct. It can be examined by insistence on high reliability of the questions, composite reliability of the constructs, and consideration of the variance extracted by constructs (AVE) (Fornell & Larcker, 1981). AVE, proposed by Fornell and Larcker (1981), considers the variance captured by the indicators. If the AVE is less than 0.5, the variance captured by the construct is less than the measurement error and the validity of the single indicator and construct is questionable. Construct reliability is demonstrated by Cronbach’s alpha. Composite reliability of constructs is calculated by squaring the sum of the loadings then dividing it by the sum of the squared loadings plus the sum of the error terms (Werts, Linn, & Joreskog, 1974).

For convergent validity, the variance extracted for each construct must be larger than 0.5, and the item-construct correlations must all be more than 0.7. All these showed that the measurement had high convergent validity. Composite reliability of each construct was also above 0.7, which is acceptable. Except reward interdependence, the Cronbach alpha of each contruct was also above 0.7, which indicated high internal consistency.

Discriminant validity determines whether the measures of the constructs are different from each other (Messick, 1980). It can be assessed by testing whether the square root of AVE is larger than correlation coefficients. Loading values for each indicator (shown in bold font in Table 5) exceeded 0.7, which indicated high and significant discriminant validity. Another way to determine it is to verify the factor loading of indicators (Chin, 1998). To have discriminant validity, indicators should have higher loading in the corresponding constructs than in other constructs. Because PLS graphs only provide factor loadings on one construct, procedures suggested by Smith, Keil & Depledge (2001) were used to generate cross-loading values, as shown in Table 5. Loading values for each indicator (shown in bold font) exceeded 0.7, which indicated high and significant discriminant validity. The discriminant validity was also assured because (1) the cross-loading table showed that all indicators had higher loading

Table 5: Cross factor loadings in Study I

	Goal	Resource	Reward					Product
	Interdependence	Interdependence	Interdependence	Task Coordination	Filtering	Molding	Mapping	Flexibility
F1	.341	.129	.215	.096	**.823**	.269	.183	.181
F2	.352	.172	.249	.134	**.868**	.205	.224	.044
M1	-.157	-.049	.042	.535	.167	**.616**	.150	-.022
M2	-.112	.043	.007	.566	.219	**.775**	.112	.243
M3	-.226	-.176	.147	.407	.182	**.614**	.354	.068
M4	-.353	-.132	.060	.646	.135	**.830**	.361	.176
MP1	-.243	-.045	-.130	.331	.046	.155	**.602**	-.245
MP2	.061	.021	.059	.288	.129	.176	**.557**	-.089
MP3	.026	-.101	.328	.157	.121	.163	**.545**	-.190
MP4	-.215	-.041	-.049	.340	.186	.495	**.662**	.117
TC1	-.165	-.090	.085	**.837**	.025	.673	.378	.133
TC2	-.075	-.130	-.078	**.708**	-.064	.518	.241	-.188
TC3	.055	.083	.058	**.783**	.178	.465	.247	.217
GD1	**.916**	.140	.237	-.116	.262	-.286	-.159	.074
GD2	**.932**	.195	.284	-.176	.245	-.280	-.172	.174
RW1	.412	-.023	**.806**	-.062	.133	-.110	.033	-.145
RW2	.450	.002	**.829**	.074	.228	-.023	-.022	.170
RS1	.406	**.757**	.234	.170	.217	.096	.097	.038
RS2	.295	**.844**	.120	.178	.307	.234	.238	-.063
RS3	.275	**.829**	.199	.099	.345	.072	.021	.171
PF1	.096	.008	.068	.123	-.001	.002	-.198	**.732**
PF2	-.005	-.095	.027	.030	.049	.092	-.305	**.815**
PF3	-.012	-.062	.143	-.074	.085	-.023	-.210	**.723**

in the corresponding constructs than in others, and (2) the square root of AVE was larger than the correlation between constructs (shown in bold font in Table 6).

2.6.4 Structural Model

The basic information about each variable is given in Table 6 including means, standard deviation, skewness and kurtosis. For each variable, the skewness was less than 2 and the kurtosis was less than 5, indicating no significant violation of normal distribution (Ghiselli, Campbell, & Zedeck, 1981).

The hierarchical moderation test is used to analyze the moderating effect. Following the suggestion from Carte and Russell (2003), the moderating effect can be assured by comparing the difference between the main effect model and the moderating effect model. This hierarchical process was adopted by many IS researchers (Gefen, Straub, & Boudreau, 2000; Khalifa & Cheng, 2002; Limayem, Hirt, & Chin, 2001; Mathieson, Peacock, & Chin, 2001). A four-step analysis process was used in this research.

I first obtained the R-square (R_1^2) of the main effect model which includes independent variable (IV, i.e. interdependence), moderator (i.e., external communication activity), and dependent variable (DV, i.e., product flexibility) only. Then the R-square (R_2^2) of the moderating effect model was obtained by including IV, moderator, interaction term (i.e., the interaction of external communication activity and interdependence), and dependent variable in the model. The interaction term used in the model is calculated by adding the multiplying result between each indicator in independent variable and each indicator in moderator (Chin, 2003). Third, I derived an estimated effect size of f^2 from (1- R_2^2) / (R_2^2 - R_1^2) and then

Table 6: Descriptive Statistics in Study I

	Mean	**Std.**	**Skewness**	**Kurtosis**	**GI**	**RI**	**RWI**	**Map**	**Filter**	**Mold**	**TC**	**PF**
Goal Interdependence	3.87	0.99	-0.96	0.49	**0.96**							
Resource Interdependence	3.46	0.85	-0.49	0.61	0.42	**0.85**						
Reward Interdependence	3.79	0.76	-0.32	-0.65	0.55	0.23	**0.86**					
Mapping	3.55	0.88	-0.51	0.46	-0.16	0.23	-0.02	**0.84**				
Filtering	3.61	0.74	-0.24	0.36	0.33	0.38	0.25	0.22	**0.92**			
Molding	3.28	0.85	-0.58	0.32	-0.20	0.21	-0.02	0.48	0.33	**0.77**		
Task Coordination	3.39	0.91	-0.55	0.03	-0.09	0.21	0.04	0.46	0.17	0.75	**0.81**	
Product Flexibility	3.77	0.67	-1.16	2.73	0.26	0.19	0.40	0.32	0.42	0.33	0.31	**0.82**

Table 7: The overall hypotheses results of Study I

R-square change	Molding	Filtering	Mapping	Task coordination
Goal interdependence	0.082*	0.098*	0.017	-
Reward interdependence	0.016	0.06*	0.041	-
Resource interdependence	0.063*	-	0.054*	0.014

obtained pseudo F-value by multiplying f^2 with $(n-k-1)$ where n is the sample size and k is the number of independent variable in the regression equation. f^2 score of 0.03, 0.15, and 0.35 imply small, moderate, and large interaction effects (Cohen, 1988). Finally, I compare the pseudo F-value with F1, n-k-1. The above four steps can test the change of variance extracted by adding a new variable (the interaction term) into the model.

Table 7 shows the results of moderation effects for each hypothesis. The molding and filtering activities had significant moderating effects on the relationship between goal interdependence and product flexibility. The filtering activity moderated the relationship between reward interdependence and product flexibility. For resource interdependence, the molding and mapping activities moderated its impacts on product flexibility. However with the constraints of a small sample size, the moderating effects on reward interdependence and resource interdependence only had significant R square changes but not significant path coefficients of interaction terms..

The hierarchical moderation results indicated that both goal interdependence and the external communication had positive impacts on product flexibility (See Table 8 in 2.9 Appendix). In addition, the changes of effect size after adding the interaction term were significant for the moderating effects of the molding and filtering activities. Therefore H1 and H3 are supported.

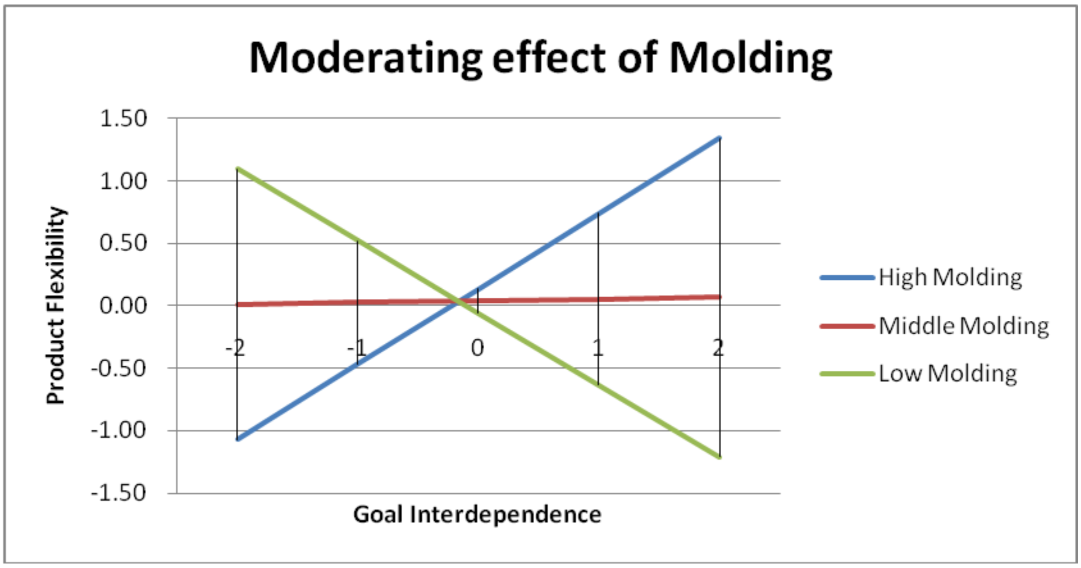


Figure 2: The moderating effect of molding on goal interdependence

To better illustrate the moderating effect of the molding activity on goal interdependence, I graphed the interaction effects following the procedures set forth by Cohen and Cohen (1987). Figure 2 shows the moderating effect of molding on the relationship between goal interdependence and product flexibility. While R square change in product flexibility was significant, it is not enough to simply assume that the interaction graph demonstrates that the change in performance is significantly different than zero without testing for the significance of the slope (Aiken & West, 1991). The path coefficient of the interaction term of molding, 0.391, was significant. Specifically, the slope significance test demonstrated that when the level of molding activity is low, the relationship between goal interdependence and program performance is significant because the coefficient is significantly different from zero. When goal interdependence is low, the low level of molding has a higher impact on product flexibility than the high level of molding. In contrast, when goal interdependence is high, the high level of molding will more likely to lead to high product flexibility than the low level of

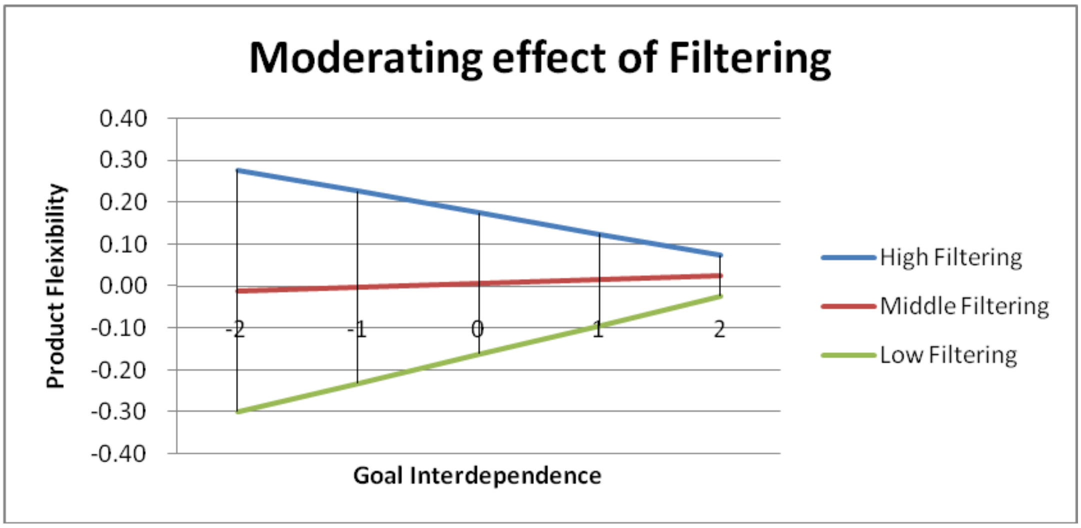


Figure 3: The moderating effect of filtering on goal interdependence

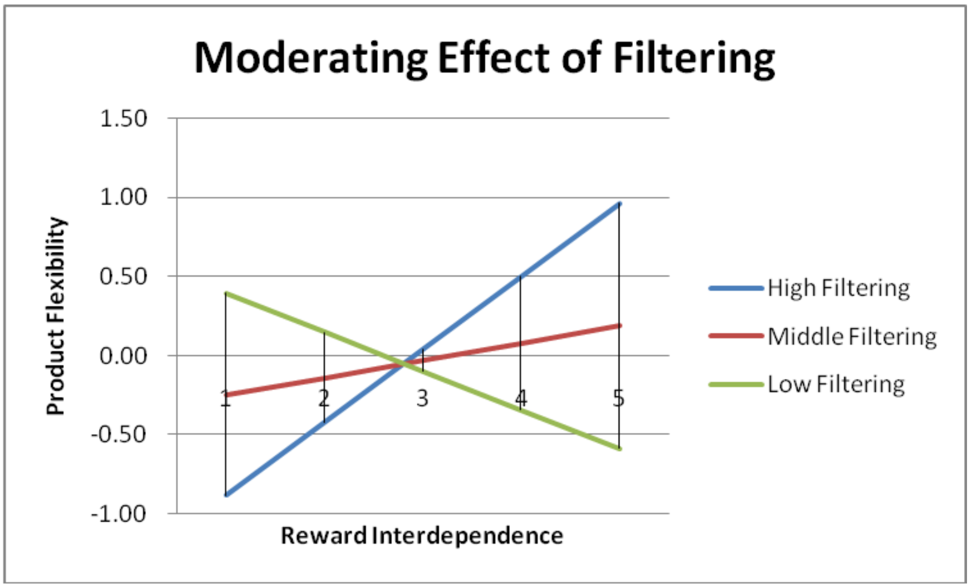


Figure 4: The moderating effect of filtering on reward interdependence

molding.

H3 states the moderating effect of filtering on goal interdependence. Figure 3 is graphed to show the interaction. When goal interdependence is low, a high level of filtering has more positive impacts on product flexibility than a low level of filtering. As the level of goal interdependence increases, the effect of a high level of filtering is decreasing but the effect of a low level of filtering is increasing.

The hierarchical moderation results showed that reward interdependence had negative impacts on product flexibility (See Table 9 in 2.9 Appendix). The change of effect size after adding the interaction term is significant for the moderating effects of filtering. Therefore H4 which states the moderating effect of molding activity on reward interdependence is not supported. The R square change for the moderating effect of mapping activity on goal interdependence is not significant either. Consequently H5 is not supported.

Figure 4 shows the moderating effect of filtering on reward interdependence. When reward interdependence is low, the low level of filtering has a higher impact on product flexibility than the high level of filtering. In contrast, when reward interdependence is high, the high level of filtering has more impacts on product flexibility than the low level of filtering.

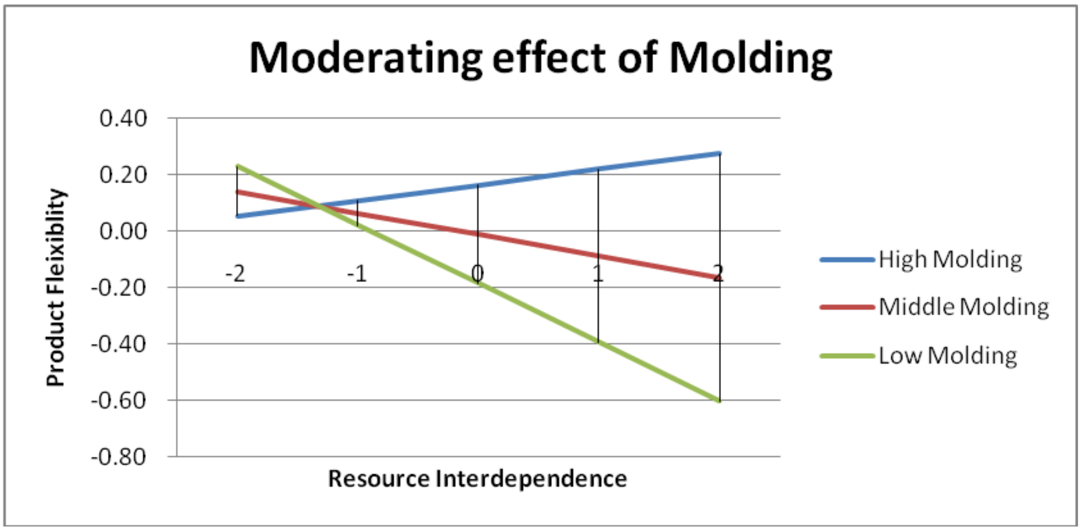


Figure 5 The moderating effect of molding on resource interdependence

The hierarchical moderation results supported that resource interdependence had negative influences on product flexibility. The change of effect size after adding the interaction term was significant for the moderating effects of mapping and molding. Therefore H9 which states the moderating effect of filtering activity on resource interdependence is not supported. H7 and H8 are supported because of significant R square changes after adding the interactions.

Figure 5 shows the moderating effect of molding on resource interdependence. When resource interdependence is low, the low level of molding has a higher impact on product flexibility than the high level of molding. In contrast, when resource interdependence is high, the high level of molding has much more impacts on product flexibility than the low level of molding.

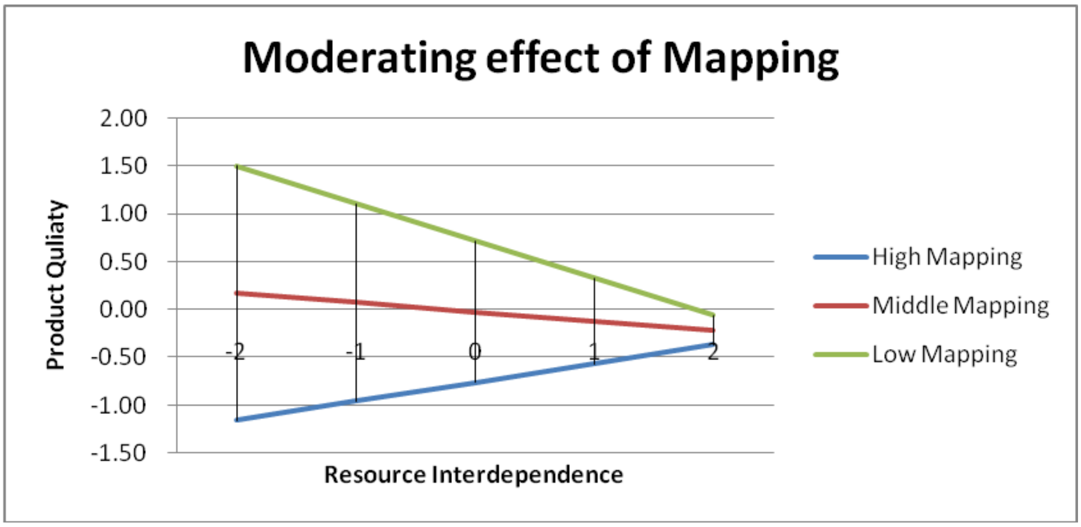


Figure 6: The moderating effect of mapping on resource interdependence

Figure 6 shows the moderating effect of mapping on resource interdependence (See Table 10 in 2.9 Appendix). When resource interdependence is low, the low level of mapping has a higher impact on product flexibility than the high level of mapping. However, when resource interdependence is high, the low level of molding still has more impacts on product flexibility than the high level of molding, but, to a less extent.

2.7 Discussion

This research examined the effects of external communication on the product development program's interdependence with the larger organization environment. The proposed framework that guided this research was based upon Resource Dependence Theory and Information Processing Theory. Specifically, the moderating effects of four external communication activities of the program management teams on interdependence were tested. As expected, both the molding and filtering activities had the moderating effects on goal interdependence. Only the filtering activity moderated the relationship between reward interdependence and the program performance. In addition, both the molding and mapping activities had the moderating effects on resource interdependence.

The further classification of the ambassadorial strategy into three sub-activities brings a lot of new insights. Out of expectations filtering is viewed very important by program managers. Filtering generates critical moderating effects on goal interdependence and reward interdependence. Figure 3 indicates that a high level of filtering has the stronger influence on the relationship between goal interdependence and product flexibility than a low level of filtering. But as goal interdependence increases, the effect of a high level of filtering is decreasing. In contrast, as goal interdependence increases, the effect of a low level of filtering is increasing.

Reward interdependence has a negative impact on program performance because the outcomes of product development depend on the collective performance of the program and the related functional lines and other programs. Figure 4 shows that when reward interdependence is low, a low level of filtering has a greater impact on product flexibility than a high level of filtering. In contrast, when reward interdependence is high, a high level of filtering contributes more to product flexibility than a low level of filtering. This interaction is counterintuitive. However, this interaction is consistent with the issues of the inter-departmental communication barriers in the past literature (Dougherty, 1992). New product development teams in large firms have persistent problems with inter-departmental communication such as the cross functional linking between marketing and technological departments. Dougherty (1992) focused on the shared interpretive schemes people use to make sense of product innovation and found that departmental thought worlds and organizational product routines inhibit the inter-departmental communication. Different frames of mind and professional backgrounds decrease the communication effectiveness. When reward interdependence is high, related functional lines are eager to get involved with the product development process. But the program management team has to selectively pass the information to the project teams and make sure the project plan can be executed successfully within the time framework.

In addition, the concept “planned isolation” in a temporary organization(Lundin & Soderholm, 1995) can deepen the understanding of the filtering activity. Projects are often viewed as a temporary organization in the organizational structure. Lundin and Soderholm (1995) argued that projects are employed more and more to deploy the organizational changes and have suggested “planned isolation” as one of the key concepts in the design of a project’s structure. Planned isolation focuses on the execution phase in the life of the projects. The minimization of any disturbance to plans or other threats to the action imperative is achieved by deliberately isolating the projects. By isolating the product development program from disturbances in the environment, the interaction between the product development program and other business functions will decrease and the development will be facilitated. In addition, another important effect of isolation is to minimize the requests from the related business functions. When goal interdependence is high, other business functions have high interests in the updates of product development. The stakeholders often make additional requests or change the requests as their interests change. Program managers have to protect the program from extra external pressure and filter the noise, building a protective layer for the projects in the program.

Lundin and Soderholm (1995) suggest two general ways of achieving isolation, planning and guarding. The program team can adopt these two strategies to achieve an appropriate extent of isolation. When product development has been put into action, program managers should stick to the action plans and use it to support the filtering of extra external pressure. In addition, program managers should try to keep the related stakeholders to stick to the original agreements, secure the path outlined by the plans, and to keep control over any changes that have been made. Of course sometimes some changes have to be made. It is the decisions that program managers should make with a great extent of caution. This planning and guiding is consistent with the concept of “consonance” proposed by Klein and Jiang (2001). The consonance concept argues for an agreement among all the stakeholders. Only when the consonance is achieved, the performance can be evaluated against the original agreement fairly.

The molding activity moderated the relationship between goal interdependence and product quality and the relationship between resource interdependence and program performance. The program manager attempts to influence the other business functions and related stakeholders to suit the product development’s agenda by shaping the beliefs and behaviors of outsiders. Program managers have to spend efforts in communicating with the senior managers from other business functions and persuade them that the product development program is important and contributes to the organizational business. The program’s goals are the sub-goals of the organizational initiatives and are aligned with the organizational business strategy. Persuading and influencing are critical when the level of resource interdependence is high. The bright image and positive light that the program manager presents to other stakeholders can attract attention from other business functions and cheer for more resources. Persuading and influencing are more likely to influence other outside stakeholders when the other business functions and the program share the same goals than when they have the conflicting goals. Some tactics that product development program manager can do include gathering information of the feasibility of product development, communicating with the business managers of the product plan, assessing the changes and implications for the business managers, and justifying and championing for the new product. This continuous communication with other business functions enables the external support for product development in terms of satisfaction with the products, feedbacks and supplementary supports for the products and future market predictions. Product quality can be continuously improved based upon the feedbacks. In addition, the updates of the product development process will keep other business functions in a loop. Their expectations can be managed towards the final product delivery.

The mapping activity is performed by the program team informally and informally. Figure 6 confirms the moderating effect of mapping on resource interdependence. But the mapping activity has a high cost for the program team. Therefore it generated negative impacts on the program performance. When the level of resource interdependence is low, a high level of the mapping activity will be a burden for the program team members and negatively influence the program performance. But when the level of resource interdependence is high, the mapping activity becomes necessary. Consequently, the cost of mapping can be justified. Except the mapping activity, the task coordinator activity is always needed for resource interdependence. But the task coordinator activity had no moderating effect on resource interdependence.

2.7.1 Implications

The past literature has studied the product development processes with the focus on communication among project team members, extensive planning and overlapping problem or experiential tactics, political and financial support and subtle control (S. L. Brown & Eisenhardt, 1995). These processors are found to improve the final outcomes. In terms of integrative structure on product development, integration of different internal sources of technological knowledge drives the product development outcomes (Iansiti, 1997). In addition, internal integration between R&D and marketing such as incentives and reward systems can positively impact the product development (Griffin & Hauser, 1996). However the interactions between the agents and these organizational processes, systems and structures have been discussed.

This study contributes to the literature by including software development characteristics in the communication process. This study will help the researchers and managers to understand in-depth the influential communication efforts and highlight the political and information-processing dynamics underlying the communication processes of successful software product development. The successful external communication will facilitate the planning and decision making in product development.

Secondly, this study improves the understandings of the role of program management team in product development. The program's success is directly linked with the firm's competition in the market. Senior managers and program managers will pick up several important implications of program operation.

Programs have unique advantages that multiple projects that are simply grouped together cannot achieve. The benefits of programs only can be generated when an effective program management team is in place. While the synchronization of multiple projects' schedules and resource optimizations are the original starting point of developing a program, the program team's bigger representation and bargaining power in the organization by external communication contribute greatly to the alignment of strategic goals and program goals, and the inter-unit integration. Therefore the capability of program management leads to sustainable competitive advantages for the organizations.

A program team's capability of external communication comes from the program manager, the composition of the program management team and their individual communication capabilities and personal networks. Senior managers should take implications from this research by selecting the candidates of program managers who understand the relationships between the program and the larger organization context and have strong capabilities in the molding, filtering and mapping activities. In addition, each program team member should be selected with considerations too. The team members' communication capability and relationships with the related departments contribute to the effectiveness of the program team's external communication. The program team should be adaptive and be good at deciding the extent of external communication according to the changes of interdependence.

Program managers should have a good personal network in the organization so that mapping and molding can be more effective. Usually a good personal network is developed because of the program manager’s personality and the past work history. In addition, program managers should be experienced in the interdepartmental communication and understand the communication barriers. The filtering activity plays a critical role in building a protective layer for the individual projects in the program. Program managers should collect the information from the related departments and stakeholders, interpret them and pass selective information to the development teams.

Although the program’s interdependence with the larger organizational context is inherent and do not change much, the program team strives to acquire critical resources for product development by external communication. The program team should take implications from this research and select the appropriate communication for different types of interdependencies.

Goal interdependence has a positive effect on program performance. However, it does not mean that the program team does nothing when goal interdependence is high. The program team has to carefully manage the program’s images and reputations in the organization and strengthen the positive impacts of goal interdependence by the molding activity. On the other hand, filtering is always necessary even though goal interdependence is high because of the different frames of mind and the consistency issue of product development.

Reward interdependence has a negative impact on program performance since program outcomes become the results of collective performances of the program and related departments. The program team cannot change the rewards by the molding or mapping activities. The only thing under the program team’s control for reward interdependence is filtering. Filtering becomes more critical when reward interdependence is high. The program team strives to maintain the product development plan and being responsive to the changes at the same time by negotiating with the other departments. The program buffers the pressures of changes and interdependence, and makes a plan for the product development teams to change at a slow pace.

Resource interdependence does not promote cooperation (Ortiz et al., 1996). It pushes the program team to actively conduct external communication for resource acquisition. The molding activity enables the program team to change the perceptions of related functional lines and motivate them to provide the resources. The mapping activity constructs a map of the possible sources of resources. However, the program team has to be cautious because of the high costs of the mapping activity.

2.7.2 Limitations and Future Research

This research examined the external communication of a product development program. It contributes to the literature by explaining why program management has the advantages that cannot be extracted by the management of individual projects and how the program team conducts extensive communication for different types of interdependences with the large organization. Both the researchers and practitioners can take away valuable insights from the interactions between external communication and interdependence.

This study has several limitations. The first limitation is the small sample size. When sample size is small, the statistical power might be sufficient (Marcoulides & Saunders, 2006). The second limitation is that the data was collection in a single country. The generalization of results has to be limited. The third limitation is that each type of interdependence is examined individually. In reality, different types of interdependencies intertwine and create complicated impacts on the program. Future research can examine the effects of interacted multiple interdependencies on program performance. At the group level, goal and resource interdependence may not elicit equivalent levels of cooperation and may even affect productivity differentially (Johnson et al., 1989). It will be interesting to explore the effects of multiple interdependencies at a program level.

Since the program team plays a critical role in program management, future research can explore the program team's competencies and how the competencies are developed and their effects on the program's outcomes. In addition, another direction that future research can consider is to study the effects of uncertainty and complexity on program performance. Interdependence can be changed as the product market shifts and as the product complexity increases. It will be interesting to see how the program team handles the effects of uncertainty and complexity, and the program's effectiveness compared with a single project team.

2.8 Conclusion

Team interface management has been identified as a critical factor for the team’s success in the large scale product development (Hoegl & Weinkauf, 2005). The inherent interdependence in the software development program requires the program manager to analyze the relationship with other business functions. This study creates a simple framework to help program managers analyze the external environment. The different types of interdependencies can give program managers a good starting point to manage the environment. Program managers should not only focus on the information exchange and technical issues but also include the activities that shape the beliefs and expectations on the product development. Several best communication strategies for each type of interdependence are supported in this study.

2.9 Appendix for Chapter Two

Table 8: Hierarchical moderation results for goal interdependence

DV: Product Flexibility (n=53)					
Path Coefficient	Model 0	Model 1	Model 2	Model 3	Model4
Goal Interdependence	0.324	0.221	0.185	0.085	0.218
Molding activities		0.264	0.293	0.334	0.261
Filtering activities		0.167	0.092	0.047	0.157
Mapping activities		-0.385	-0.322	-0.245	-0.374
Interaction1 (Molding x GI)				0.391*	
Interaction2 (Filtering x GI)					-0.112
Interaction3 (Mapping x GI)			0.317*		
R2	0.105	0.343	0.425	0.441	0.36
R2 Change		0.238*	0.082*	0.098*	0.017

Table 9: Hierarchical moderation results for reward interdependence

DV: Product Flexibility (n=53)					
Path Coefficient	Model 0	Model 1	Model 2	Model 3	Model4
Reward Interdependence	-0.327	-0.229	-0.221	-0.18	-0.261
Molding activities		0.211	0.21	0.235	0.23
Filtering activities		0.222	0.197	0.129	0.107
Mapping activities		-0.418	-0.388	-0.381	-0.371
Interaction1 (Molding x RWI)				0.233	
Interaction2 (Filtering x RWI)					0.2
Interaction3 (Mapping x RWI)			0.12		
R2	0.107	0.353	0.366	0.394	0.413
R2 Change		0.246*	0.013	0.041	0.06*

Table 10: Hierarchical moderation results for resource interdependence

	DV: Product Flexibility (n=53)				
Path Coefficient	Model 0	Model 1	Model 3	Model 2	Model4
Resource Interdependence	-0.308	-0.121	-0.187	-0.181	-0.129
Molding activities		0.266	0.306	0.307	0.274
Filtering activities		0.208	0.079	0.09	0.196
Mapping activities		-0.416	-0.295	-0.378	-0.401
Interaction1 (Molding x RI)			-0.304		
Interaction2 (Filtering x RI)					-0.073
Interaction3 (Mapping x RI)				0.267	
R2	0.095	0.322	0.385	0.376	0.33
R2 Change		0.227*	0.063*	0.054*	0.008

Table 11: Survey items used in Study I

Constructs	Labels	Items
Filtering	F1	absorb outside pressures for the program
	F2	protect the projects in this program from outside interference
Molding	M1	scan the environment in the organization for threats
	M2	help business units to know this program
	M3	find out strategy information or political situation that may affect the program
	M4	keep business units informed of this program's activities
Mapping	MP1	persuade business units that the program is important
	MP2	acquire resources for the program
	MP3	report this program's progress to senior managers
	MP4	find out whether business units support the program's activities
Task coordination activity	TC1	resolve design problems with business units
	TC2	coordinate activities with business units
	TC3	review system design with business units
Goal interdependence	GD1	program goals come from clients' goals.
	GD2	program goals are determined by clients' goals.
Reward interdependence	RW1	Feedback about program performance comes from clients' evaluation of products.
	RW2	program performance evaluations are strongly influenced by clients.
Resource interdependence	RS1	the extent of sharing testing data
	RS2	the extent of sharing expertise
	RS3	the extent of sharing product information
Product flexibility	PF1	products are adapted to changes with cost efficiency.
	PF2	products are rapidly adapted to changes.
	PF3	products have long term flexibility.

2.10 Chapter Two List of References

Aiken, L. S., & West, S. G. (1991). *Multiple Regression: Testing and Interpreting Interactions*. Thousand Oaks, CA.: Sage.

Ancona, & Caldwell, D. F. (1992). Bridging the Boundary: External Activity and Performance in Organizational Teams. *Administrative Science Quarterly, 37*(4), 634-661.

Andres, H. P., & Zmud, R. W. (2001). A Contingency Approach to Software Project Coordination. *Journal of Management Information Systems, 18*(3), 41.

Boehm, B., W. . (1991). Software Risk Management: Principles and Practices. *IEEE Software, 8*(1), 32.

Brown, M. M., J. O'Toole, L., Jr., & Brudney, J., L. . (1998). Implementing information technology in government: An empirical assessment of the role of local partnerships. *Journal of Public Administration Research and Theory, 8*(4), 499.

Brown, S. L., & Eisenhardt, K. M. (1995). Product Development: Past Research, Present Findings, and Future Directions. *Academy of Management Review, 20*(2), 343-378.

Brusoni, S., & Prencipe, A. (2001). Unpacking the black box of modularity: Technologies, products and organizations. *Industrial and Corporate Change, 10*(1), 179.

Brusoni, S., Prencipe, A., & Pavitt, K. (2001). Knowledge specialization, organization coupling, and the boundries of the firm: Why do firms know more than they make? *Administrative Science Quarterly, 46*(4), 597.

Burgelman, R. A., Maidique, M. A., & Wheelwright, S. C. (1996). *Strategic management of technology and innovation*. Chicago: Irwin.

Campion, M. A., Medsker, G. J., & Higgs, A. C. (1993). Relations Between Work Team Characteristics and Effectiveness: Implications for Designing Effective Work Groups. *Personnel Psychology, 46*(4), 823-850.

Campion, M. A., Papper, E. M., & Medsker, G. J. (1996). Relations Between Work Team Characteristics and Effectiveness: A Replication and Extension. *Personnel Psychology, 49*(2), 429-452.

Carte, T. A., & Russell, C. J. (2003). In Pursuit of Moderation: Nine Common Errors and Their Solutions. *MIS Quarterly, 27*(3), 479-501.

Chin, W. W. (1998). *The partial least squares approach to structural equation modeling*. New Jersey: Lawrence Erlbaum Associates.

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, New Jersey: Lawrence Erlbaum.

Cohen, J., & Cohen, P. (1987). *Applied Multiple Regression/Correlation Analyses for The Behavioral Sciences* (3rd. ed.). Hillsdale, NJ.: Lawrence Erlbaum.

Cooprider, J. G., & Henderson, J. C. (1990). Technology-Process Fit: Perspectives on Achieving Prototyping Effectiveness. *Journal of Management Information Systems, 7*(3), 67-87.

Cossick, K. L., Byrd, T. A., & Zmud, R. W. (1992). A Synthesis of Research on Requirements Analysis and Knowledge Acquisition Techniques. *MIS Quarterly, 16*(1), 117-138.

Daft, R. L., & Lengel, R. H. (1986). Organizational Information Requirements, Media Richness and Structural

Design. *Management Science, 32*(5), 554-571.

Dougherty, D. (1992). Interpretive Barriers to Successful Product Innovation in Large Firms. *Organization Science, 3*(2), 179-202.

Driskill, L. P., & Goldstein, J. R. (1986). Uncertainty: Theory and Practice in Organizational Communication. *Journal of Business Communication, 23*(3), 41-56.

Duncan, N. B. (1995). Capturing Flexibility of Information Technology Infrastructure: A Study of Resource Characteristics and their Measure. *Journal of Management Information Systems, 12*(2), 37-57.

Fan, E. T., & Gruenfeld, D. H. (1998). When Needs Outweigh Desires: The Effects of Resource Interdependence and Reward Interdependence on Group Problem Solving. *Basic & Applied Social Psychology, 20*(1), 45-56.

Faraj, S., & Sproull, L. (2000). Coordinating Expertise in Software Development Teams. *Management Science, 46*(12), 1554.

Floyd, S. W., & Wooldridge, B. (1992). Managing strategic consensus: the foundation of effective implementation. *Academy of Management Executive*

Academy of Management Executive J1 - Academy of Management Executive, 6(4), 27-39.

Fornell, C., & Larcker, D. F. (1981). Evaluating Structural Equation Models with Unobservable Variables and Measurement Error. *Journal of Marketing Research (JMR), 18*(1), 39-50.

Galbraith, J. R. (1973). *Designing Complex Organizations*. Reading, Mass.: Addison-Wesley Pub. Co.

Gales, L., Porter, P., & Mansour-Cole, D. (1992). Innovation project technology, information processing and performance: A test of the Daft and Lengel conceptualization. *Journal of Engineering and Technology Management, 9*(3-4), 303-338.

Gattiker, T. F., & Goodhue, D. L. (2005a). What Happens After ERP Implementation: Understanding the Impact of Interdependence and Differentiation on Plant-Level Outcomes. *MIS Quarterly, 29*(3), 559-585.

Gefen, D., Straub, D. W., & Boudreau, M. (2000). Structural equation modeling and regression: Guidelines for research practice. *Communications of the AIS, 7*, 1-78.

Gerwin, D., & Moffat, L. (1997). Authorizing processes changing team autonomy during new product development. *Journal of Engineering & Technology Management, 14*(3/4), 291.

Ghiselli, E., Campbell, J., & Zedeck, S. (1981). *Measurement theory for the behavioral sciences*. San Francisco: WH Freeman.

Gladstein, D. L. (1984). Groups in Context: A Model of Task Group Effectiveness. *Administrative Science Quarterly, 29*(4), 499-517.

Griffin, A., & Hauser, J. R. (1996). Integrating R&D and marketing: A review and analysis of the literature. *Journal of Product Innovation Management, 13*(3), 191-215.

Henderson, R. M., & Clark, K. B. (1990). Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms. *Administrative Science Quarterly, 35*(1), 9-30.

Hoegl, M., & Weinkauf, K. (2005). Managing Task Interdependencies in Multi-Team Projects: A Longitudinal Study. *Journal of Management Studies, 42*(6), 1287-1308.

Hoegl, M., Weinkauf, K., & Gemuenden, H. G. (2004). Interteam Coordination, Project Commitment, and Teamwork in Multiteam R&D Projects: A Longitudinal Study. *Organization Science, 15*(1), 38-55.

Hulland, J. (1999). Use of partial least squares (PLS) in strategic management research: a review of four recent studies. *Strategic Management Journal, 20*(2), 195-204.

Iansiti, M. (1997). From technological potential to product performance: an empirical analysis. *Research Policy, 26*(3), 345-365.

Jalote, P. (2000). *CMM in Practice: Processes for Executing Software Projects at Infosys*. Boston, MA.: Addison-Wesley

Jensen, C., Johansson, S., & Lofstrom, M. (2006). Project relationships - A model for analyzing interactional uncertainty. *International Journal of Project Management, 24*(1), 4-12.

Johnson, D. W., Johnson, R. T., & Stanne, M. B. (1989). Impact of Goal and Resource Interdependence on Problem-Solving Success. *Journal of Social Psychology, 129*(5), 621.

Kazanjian, R. K., Drazin, R., & Glynn, M. A. (2000). Creativity and technological learning: the roles of organization architecture and crisis in large-scale projects. *Journal of Engineering & Technology Management, 17*(3/4), 273.

Khalifa, M., & Cheng, S. (2002). *Adoption of Mobile Commerce: Role of Exposure* Paper presented at the 35th Annual Hawaii International Conference on System Sciences (HICSS'02) Hawaii.

Kim, K. K., Umanath, N. S., & Kim, B. H. (2005). An Assessment of Electronic Information Transfer in B2B Supply-Channel Relationships. *Journal of Management Information Systems, 22*(3), 293-320.

Klein, G., & Jiang, J. J. (2001). Seeking consonance in information systems. *Journal of Systems and Software, 56*(2), 195-202.

Kraut, R. E., & Streeter, L. (1995). Coordination in Software Development. *Communications of the ACM, 38*(3), 69-81.

Krishnan, M. S., Kriebel, C. H., Kekre, S., & Mukhopadhyay, T. (2000). An Empirical Analysis of Productivity and Quality in Software Products. *Management Science, 46*(6), 745.

Lee, G., & Xia, W. (2005). The ability of information systems development project teams to respond to business and technology changes: a study of flexibility measures. *European Journal of Information Systems, 14*(1), 75.

Leifer, R., & Delbecq, A. (1978). Organizational/Environmental Interchange: A Model of Boundary Spanning Activity. *Academy of Management Review, 3*(1), 40-50.

Limayem, M., Hirt, S. B., & Chin, W. W. (2001, June 27-29). *Intention does ont always matter: The contingent role of habit on IT usage behavior.* Paper presented at the The 9th European Conference on Information Systems, Bled, Slovenia.

Locke, E. A., & Latham, G. P. (1990). *A theory of goal setting & task performance* Englewood Cliffs, N.J. : Prentice Hall.

Lohmoller, J.-B. (1989). *Latent Variable Path Modeling with Partial Least Squares* Heidelberg: Physica-Verlag.

Lundin, R. A., & Soderholm, A. (1995). A theory of the temporary organization. *Scandinavian Journal of Management, 11*(4), 437-455.

Lycett, M., Rassau, A., & Danson, J. (2004). Programme management: a critical review. *International Journal of Project Management, 22*(4), 289.

Marcoulides, G. A., & Saunders, C. (2006). PLS: A Silver Bullet? *MIS Quarterly,* pp. 211-211, from http://ucfproxy.fcla.edu/login?URL=http://search.ebscohost.com/login.aspx?direct=true&db=buh

&AN=21145593&site=ehost-live

Mathieson, K., Peacock, E., & Chin, W., W. . (2001). Extending the technology acceptance model: The influence of perceived user resources. *Database for Advances in Information Systems, 32*(3), 86.

McFarlan, F. W. (1989). Portfolio approach to information systems. In *Software risk management* (pp. 17-25): IEEE Press.

Messick, S. (1980). Test validity and the ethics of assessment. *American Psychologist, 35*(11), 1012-1027.

Nidumolu. (1995). The Effect of Coordination and Uncertainty on Software Project Performance: Residual Performance Risk as an Intervening Variable. *Information Systems Research, 6*(3), 191-219.

Nidumolu, S. R. (1996). A comparison of the structural contingency and risk-based perspectives on coordination in... *Journal of Management Information Systems, 13*(2), 77.

Olson, E. M., Walker, J. O. C., & Ruekert, R. W. (1995). Organizing for effective new product development: The moderating role of product innovativeness. *Journal of Marketing, 59*(1), 48.

Ortiz, A. E., Johnson, D. W., & Johnson, R. T. (1996). The Effect of Positive Goal and Resource Interdependence on Individual Performance. *Journal of Social Psychology, 136*(2), 243-249.

Partington, D., Pellegrinelli, S., & Young, M. (2005). Attributes and levels of programme management competence: an interpretive study. *International Journal of Project Management, 23*(2), 87-95.

Pearce, J. L., & Gregersen, H. B. (1991). Task Interdependence and Extrarole Behavior: A Test of the Mediating Effects of Felt Responsibility. *Journal of Applied Psychology, 76*(6), 838-844.

Pellegrinelli, S. (1997). Programme management: organising project-based change. *International Journal of Project Management, 15*(3), 141-149.

Pellegrinelli, S. (2002). Shaping context: the role and challenge for programmes. *International Journal of Project Management, 20*(3), 229.

Pellegrinelli, S. (2004). The difference between project and programme management. *Computer Weekly*, 36.

Petter, S., Straub, D., & Rai, A. (2007). Specifying Formative Constructs in Information Systems Research. *MIS Quarterly, 31*(4), 623-656.

Pfeffer, J., & Salancik, G. R. (1978). *The external control of organizations : a resource dependence perspective* Stanford, Calif: Stanford Business Books.

Platje, A., Seidel, H., & Wadman, S. (1994). Project and portfolio planning cycle : Project-based management for the multiproject challenge. *International Journal of Project Management, 12*(2), 100-106.

Podsakoff, P. M., MacKenzie, S. B., Lee, J.-Y., & Podsakoff, N. P. (2003). Common method biases in behavioral research: A critical review of the literature and recommended remedies. *Journal of Applied Psychology, 88*(5), 879-903.

Ragatz, G. L., Handfield, R. B., & Petersen, K. J. (2002). Benefits associated with supplier integration into new product development under conditions of technology uncertainty. *Journal of Business Research, 55*(5), 389-400.

Rubinstein, D. (2007). Standish Group Report: There's Less Development Chaos Today [Electronic Version]. *SD Times*. Retrieved 05/31/2008, from http://www.sdtimes.com/content/article.aspx?ArticleID=30247

Sanchez, R., & Mahoney, J. T. (1996). Modularity, Flexibility, and Knowledge Management in Product and

Organization Design. *Strategic Management Journal, 17*(Winter Special Issue), 63.

Schmidt, R., Lyytinen, K., Keil, M., & Cule, P. (2001). Identifying Software Project Risks: An International Delphi Study. *Journal of Management Information Systems, 17*(4), 5-36.

SEI. (1994). *The Capability Maturity Model: Guidelines for Improving the Software Process*. Reading, MA: Addison-Wesley.

Sharma, R., & Yetton, P. (2003). The contingent effects of management support and task interdependence on successful information systems implementation. *MIS Quarterly, 27*(4), 533-555.

Smith, H. J., Keil, M., & Depledge, G. (2001). Keeping Mum as the Project Goes Under: Toward an Explanatory Model. *Journal of Management Information Systems, 18*(2), 189-227.

Song, M., & Montoya-Weiss, M. M. (2001). The Effect of Perceived Technological Uncertainty on Japanese New Product Development. *The Academy of Management Journal, 44*(1), 61-80.

Standish. (1995). *Standish Group Report CHAOS* o. Document Number)

Standish. (2002). Standish Group Report CHAOS.

Thiry, M. (2004). "For DAD": a programme management life-cycle process. *International Journal of Project Management, 22*(3), 245.

Tillquist, J., King, J. L., & Woo, C. (2002). A representational scheme for analyzing information technology and organizational dependency. *MIS Quarterly, 26*(2), 91-118.

Tushman, M. L., & Scanlan, T. J. (1981). Characteristics and External Orientations of Boundary Spanning Individuals. *Academy of Management Journal*

Academy of Management Journal J1 - Academy of Management Journal, 24(1), 83-98.

Ulrich, D., & Barney, J. B. (1984). Perspectives in Organizations: Resource Dependence, Efficiency, and Population. *Academy of Management Review, 9*(3), 471-481.

Van De Ven, A. H., Delbecq, A. L., & Koenig, R. (1976). Determinants of Coordination Modes Within Organizations. *American Sociological Review, 41*(2), 322-338.

Vancouver, J. B., & Schmitt, N. W. (1991). An Exploratory Examination of Person-Organization Fit: Organizational Goal Congruence. *Personnel Psychology, 44*(2), 333-352.

Verona, G. (1999). A Resource-based View of Product Development. *Academy of Management Review, 24*(1), 132-142.

von Hippel, E. (1990). Task partitioning: An innovation process variable. *Research Policy, 19*(5), 407-418.

Wageman, R. (1995). Interdependence and Group Effectiveness. *Administrative Science Quarterly, 40*(1), 145-180.

Waldman, D. A., Ramirez, G. G., House, R. J., & Puranam, P. (2001). Does Leadership Matter? CEO Leadership Attributes and Profitability under Conditions of Perceived Environmental Uncertainty. *Academy of Management Journal, 44*(1), 134-143.

Weerd, I. v. d., Brinkkemper, S., Nieuwenhuis, R., Versendaal, J., & Bijlsma, L. (2006). *Towards a reference framework for software product management* Paper presented at the 14th IEEE International Requirements Engineering Conference (RE'06).

Weick, K. E., & Roberts, K. H. (1993). Collective Mind in Organizations: Heedful Interrelating on Flight Decks. *Administrative Science Quarterly, 38*(3), 357-381.

Werts, C., Linn, R., & Joreskog, K. (1974). Intraclass Reliability Estimates: Testing Structural

Assumptions *Educational and psychological measurement, 34*(1), 25.

CHAPTER THREE:
COORDINATING MULTIPLE INTERDEPENDENT PROJECTS IN INNOVATIVE PRODUCT DEVELOPMENT PROGRAMS

3.1 Introduction

The large scale product development process is challenging because of the different project schedules, project interdependence and communication difficulties across the project boundaries. In the general product development literature, researchers have begun to investigate the large-scale product development projects and focused on selected aspects such as authorizing processes (Gerwin & Moffat, 1997), learning and creativity (Kazanjian, Drazin, & Glynn, 2000), knowledge specialization (Brusoni & Prencipe, 2001), task partitioning (von Hippel, 1990) and coordination (Hoegl, Weinkauf, & Gemuenden, 2004; Sanchez & Mahoney, 1996). However, the management of the large-scale projects can become even more challenging when the product is innovative.

The innovativeness of the product is a critical factor that influences the final product development performance (Olson, Walker, & Ruekert, 1995). The innovation projects are characterized as an unfolding process. The product technology has a part of unknown and the complex interdependency causes systematic impacts on the product and the final delivery time. In addition, the product innovativeness causes more exceptions. The exceptions can become more complicated as the product becomes complex. This increasing complexity enlarges the effects of the high level of uncertainty inherent in the development of innovative products. A larger and richer amount of information has to be processed to deal with the need of coordination (Gales, Porter, & Mansour-Cole, 1992).

The innovativeness and complexity of the large scale product development process demands a large amount of coordination and information exchange. Multiple interdependent projects are started to be employed in the process. This approach of grouping multiple projects together that share common goals is called program management (Pellegrinelli, 1997). A program is a framework for grouping existing projects or defining new projects and for focusing all the activities required to achieve a set of major benefits. Related projects in a program are managed in a coordinated way to achieve a common goal, or to extract benefits which would not be realized if they were managed independently (Pellegrinelli, 1997) ”. Although there are many types of innovative products, this study focuses on software products or the products that are integrated by software, and examines the management issues of a software product development program.

A software product development program creates benefits through better organization of projects and their activities. A software product development program is responsive to business’ needs in an uncertain competitive environment. Businesses face violate markets and a high level of technological uncertainty. Changing market needs and competitive pressures push the organization to take action. A supportive development environment is created by the program for the projects both in exploratory nature and with identified objectives. A software product development program also takes a wider view to ensure that projects’ activities will achieve the overall business benefits instead of satisfying several project clients or sponsors in the organization. Without program management, projects are competing for resources in the organization with other projects and functional units directly. Decisions are made based upon the narrow views of involved project supporters.

When uncertainty generates negative impacts on the development process, the inherent complexity of the large scale product development process doubles the negative impacts. A large amount of information and coordination are needed between the interdependent projects. A software product development program builds a small context for the multiple interdependent projects and makes the inter-project coordination easier. Based upon Information Processing Theory (IPT) (Galbraith, 1973), the extent of the inter-project coordination depends on the needs for information processing such as uncertainty and interdependence. Across all kinds of development projects, not any one type of coordinating structure is likely to be uniformly successful in delivering creative new products, cutting development time, and improving new product success in the marketplace.

Researchers have viewed business and technology changes as critical software development risks (Boehm, 1991; Schmidt, Lyytinen, Keil, & Cule, 2001). The extent of the inter-project coordination should be contingent upon the needs of information exchange because of business and technology changes. Although a few studies have examined the coordination issues in multiple projects (Hoegl & Weinkauf, 2005; Hoegl et al., 2004), the interaction between the extent of inter-project coordination and uncertainty is still unknown.

In addition, the past studies of large scale projects (Hoegl & Weinkauf, 2005; Hoegl et al., 2004) only examined the management of tangible and economic resource dependencies, which is defined as administrative coordination by Faraj and Sproull (2000). Knowledge is a type of intangible but critical resources which is crucial for non-routine intellectual team work. The management of knowledge and skill dependencies is called expertise coordination (Faraj & Sproull, 2000). Both administrative and expertise coordination are needed for the teamwork process. Expertise coordination is not only important during a single project team's work process but also plays a critical role in the multiple-project environment. In addition to the inter-project administrative coordination, the product development program performance is dependent on having the "right" expertise, creating knowledge through inter-project expertise coordination, and solving the emergent problems.

This study focuses on the management tactics that can manage uncertainty through two types of inter-project coordination: administrative coordination and expertise coordination. The research question that this paper addresses is *"How can a software development program manage the software product development uncertainties through inter-project coordination? "*

This study contributes to the literature in two ways. First, it will provide an in-depth understanding of the moderating effect of inter-project coordination on uncertainty. Requirement uncertainty and technological uncertainty are examined in the study. The coordination effectiveness on different levels of uncertainty is pinpointed. Second, this study examines expertise coordination between multiple project teams. The unbalanced distribution of experts and knowledge in different project teams create the need for bringing the expertise in when the tasks cannot be solved by the present knowledge in the team. Expertise coordination has more meaning and importance in this innovative product development process.

Literature review of uncertainty and coordination will be briefly presented in Section 3.2. A theoretical framework is proposed based upon IPT in Section 3.3. Following the theoretical framework, the hypotheses are developed in Section 3.4. Research methodology is discussed in Section 3.5. Data analysis results are presented in Section 3.6. The results are discussed in Section 3.7. At last, Section 3.8 concludes this study.

3.2 Literature Review

3.2.1 Uncertainty

According to Galbraith (1973), uncertainty can be defined as the difference between the amount of information it needs to accomplish its task in its particular environment and the amount of information it already has. Daft and Lengel (1986) argued that uncertainty includes both "lacking of information" and ambiguity which means "lacking of understanding of the tasks". Product development usually has the innovative tasks. Because of the newness of the tasks, the product development team often has to figure out a common understanding of the tasks. Therefore lacking understanding of the tasks often creates more risks for the product development process than lacking amount of information. Driskill and Goldstein (1986) propose that uncertainty is the perceived lack of information, knowledge, beliefs and feelings – whatever is necessary for accomplishing the organizational task and the personal objectives of communicators in the organization. This definition is particularly suitable in this study since this study focuses on not only the extent of the coordination but also the contents of the coordination such as knowledge and beliefs. When the change is ambiguous and multiple stakeholders have different beliefs and opinions on the change, the coordination process will be more difficult.

Software development uncertainty usually comes from the changes in the business and technological environment. Lee and Xia (2005) has analyzed the changes in the software development process and categorized the changes into business and technological changes from the socio-technical perspective. The Information Systems Development Project's (ISDP)'s business context frequently changes during the development process. These business changes subsequently result in changes in user requirements of the software system under development. Prior literature has discussed various types of business and user requirement changes (Jalote, 2000). For example, the software Capability Maturity Model (CMM) literature makes distinctions between technical user requirements and non-technical user requirements (SEI, 1994). Researchers have viewed business and technology changes as critical software development risks. For example, Boehm (1991) ranked business requirement changes as a top software project risk. Software project risks include various business and technology changes such as unstable corporate environments, changing scope/objectives, introduction of new technology, and instability of technical architecture (Schmidt et al., 2001).

Business changes are usually signaled in the competitive market first. Firms have encountered increased environmental uncertainty and rapid changes in their external environments (Waldman, Ramirez, House, & Puranam, 2001). Competitors make marketing movements frequently and aggressively and create a state of constant change. Organizations usually have to shorten product life cycle and develop short design life cycle to compete with other players in the market (S. L. Brown & Eisenhardt, 1995).

When business changes are recognized, they are translated into requirements by the program management team. Requirement uncertainty in software development processes has been widely studied because of the difficulty of eliciting requirements from users (Cossick, Byrd, & Zmud, 1992; Nidumolu, 1995). Requirement uncertainty has three dimensions: requirement instability, requirement diversity and requirement unanalyzability (see Table 12 for the definition of each dimension) (Nidumolu, 1996). A high level of requirement uncertainty will need more coordination efforts and lead to less process control and product flexibility (Nidumolu, 1996).

Table 12: Definitions of uncertainty

Uncertainties	**Dimensions**	**Definitions**
Requirement Uncertainty	Requirement Instability	the extent of change in user requirements from the early phase to the later stage of software product development.
	Requirement Diversity	the extent to which users differed amongst themselves in their requirements of a complex software product.
	Requirement Unanalyzability	the extent to which a conversion process can be reduced to mechanical steps or objective procedures.
Technology Uncertainty	Technological Unpredictability	the extent to which unexpected and novel events for the technology occur during the software development process.
	Technological Unanalyzability	the extent to which the task of converting requirements specifications to software could be undertaken using well-established procedures.
Adopted from (Nidumolu, 1996)		

Technological uncertainty is another source of threats to product development (Nidumolu, 1996). Technology is theoretical and practical knowledge, skills, and artifacts that can be used to develop products and services as well as their production and delivery systems (Burgelman, Maidique, & Wheelwright, 1996). Typically, an ISDP team deals with two types of information technologies: software development tools and infrastructure technologies (Cooprider & Henderson, 1990). Software development tools include programming languages/tools, computer-aided software engineering (CASE) tools, and packaged software. These tools are selected for specific ISDPs based on business needs and technological capabilities. In contrast, infrastructure technologies are shared by multiple ISDPs. According to Duncan (1995), infrastructure technologies include platform technology, network/telecommunication technologies, key enterprise data, and core enterprise applications.

Technology is creating new imperatives for the conduct and structuring of product development activities because new knowledge is being applied at a faster rate, greater numbers of new products are being introduced over time, the time between innovations is decreasing, and technological fusion is occurring across and within industries (Song & Montoya-Weiss, 2001). Perceived technological uncertainty has negative impacts on the costs of new product development and will result in distinct managerial actions (Ragatz, Handfield, & Petersen, 2002; Song & Montoya-Weiss, 2001). Boehm (1991) found that projects that venture into advanced technology or push the boundary of a project team's technological capability increase the risks of failure. McFarlan (1989) views the organization's experience with technology (such as hardware, operating system, database, application languages) as a key source of uncertainty. The adoption of any specific technology is a big decision for product development. However, even after the software development has been started, technological uncertainty still requires the product manager's intensive attention. Moreover, the software product industry has the fashions that customers often pursue and these customers push for the newest technology as much as possible.

3.2.2 Product knowledge, organizational structure and product integration

Knowledge is abstract representation but expertise is defined as the possession of such knowledge. Product development programs enable an organization to pool together a wide range of expertise from different business units to accomplish complex tasks and prioritize the resource allocation by ranking the strategic importance of projects. In traditional business operation, vital knowledge is often concealed by functional boundaries. In a program, expertise from different functions and in different levels can be accessed more easily.

The development of a complex product in a program requires two types of knowledge, component knowledge and architecture knowledge (Brusoni, Prencipe, & Pavitt, 2001). Product development requires component knowledge, or knowledge about each of the core design concepts and the way in which they are implemented in a particular component. It also requires architectural knowledge or knowledge about the ways in which the components are integrated and lined together into a coherent whole. The change in a component has to be carefully studied to make a judgment whether the links between the components will be affected. Component change is a change in the core design concept of a component that does affect its relationships with the others. Architectural change is often triggered by a change in a component in terms of new interactions and new linkages with other components in the system. But the main design, features or functions will be kept same. The architectural knowledge is a source of insight into the way to estimate the impacts of uncertainty on the software product development. Decisions must be made for the new changes and coordination efforts are needed to handle the changes and adapt the software system to the new changes.

However knowledge and organizational coordination cannot be achieved by relying only on automatic mechanisms enabled by the modular product architectures (Brusoni & Prencipe, 2001). Rather, the achievement of knowledge and organizational coordination demands interactive management of the actors and activities involved. This role of coordination is played by "systems integrators" or “Software Architects”. Usually the systems integrators span capabilities over a range of technological fields that is wider than the range of activities that they actually perform in-house (Brusoni et al., 2001). Their knowledge coordination will pinpoint the nature of the changes and identify the affected components and links between components. This smooth knowledge coordination will increase the likelihood of successful design change and lead to product delivery on time.

But every program usually only has one or two system integrators. These system integrators are the most precious resource in the product development program. Other types of knowledge resources that can be coordinated are the written documents from the past, product road-mapping and the overall architectural design, etc.

A product development program can be more effective in responding to requirement and technological uncertainty if the program team can manage the inter-project coordination well. The product that a program is used to develop is usually more complex than the product that a single project can handle. The innovative nature of product development requires not only the presence of experts but also the emergent expertise coordination in the program.

Expertise is context specific and emerges from patterned interactions and practices (Faraj & Sproull, 2000). Therefore it is critical for the project teams in a program to know where expertise is and when expertise is needed. The program management team should be able to understand the project teams’ need and bring the expertise to solve the problems quickly and responsively. The expertise coordination process will contribute to the problem solving for the project teams effectively and efficiently.

3.2.3 Expertise coordination processes in the software development program

Expertise coordination was originally proposed in a single project team setting. Product development programs build a context for the multiple projects that share the same goals. Product development programs need expertise coordination because of the distributed expertise in different project teams, the inherent interdependence among projects and the innovative product development.

Faraj and Sproull (2000) propose that expertise coordination consists of socially shared cognitive processes that develop and evolve in order to meet the demands of task-based skill and knowledge dependencies. When team members apply expertise to meet task demands, they activate and reinforce these processes. Expertise coordination processes require differentiated knowledge and skills possessed by team members and patterns of heedful interactions that support the application of these skills and knowledge where needed.

Product development programs have distributed experts who are dispersed in different project teams. The communication processes between experts in different project teams can be heedful of coordination since they share overlapping task knowledge and can take joint actions to handle the changes. The communication processes are emergent since there are no pre-determined answers for the new rising needs from uncertainty.

According to Faraj and Sproull (2000), expertise coordination has three dimensions:

knowing where expertise is located, recognizing where it is needed, and bringing it to bear. They are not rigid steps that must occur in a preset temporal progression. They represent general patterns of activity that a team needs to manage to be effective.

In a program, the experts in other projects team are treated as internal knowledge resources. Knowing expertise location requires knowing about a variety of potentially useful expertise sources. These sources can include specialized documents, corporate Q&A files, and most important for knowledge work, knowing who has what knowledge/skill. Only in the simplest situation does knowing expertise location refer to knowing where an answer to a problem is located. In nontrivial cases, it refers to knowing the most effective expertise to call on to develop a solution. Recognizing the need for expertise is critical for coordination. The delay of recognition of the need will leave the problems unsolved and cause schedule delay and even lengthen the time to market.

When the need for expertise is identified, the most important thing will be to bring the expertise to bear. In a single team setting, interpersonal interaction is easier to achieve and rich information can be exchanged with the experts to work on the tasks that have special needs. But in a multiple-project environment, bringing an expert to the most urgent tasks in other projects involves more than task coordination. Many times it has to be done by administrative procedures. Program managers and related project managers in the program team have to be involved in identifying the importance of need. Sometimes the experts are official assigned to solve the problems in a project team. Sometimes the experts just give some tips to the team members to have a trial and error start via an informal talk. When the information is equivocal, the interpretation will be difficult even with experts' helps. The program management team plays a critical role in understanding the essence of the problems and bringing the appropriate experts to the project teams.

3.3 Theoretical Framework

The past literature only examines the product innovation in a brief way. This study further examines the product innovativeness in two perspectives: requirement uncertainty and technological uncertainty. Product requirements can change significantly from the early phase to the later stage of product development. The requirements from each group might be very different. Sometimes the requirements even cannot be articulated by the clients clearly. The technology that is used in the product development process can change out of expectation and create crucial problems for product compatibility and integration.

A single software development team can use vertical and horizontal coordination to reduce the high level of uncertainty and lead to the project success (Nidumolu, 1996). But as the software system becomes more complex, a single project's resource is not sufficient for responding to uncertainty and changes. It is very difficult for a project team to lobby for resources in an organization. In addition, it is easier to compete against other interdependent projects within a program. The program will prioritize the project needs and coordinate among multiple projects to mitigate the negative impacts of uncertainty on the product development.

Administrative coordination has been shown in previous research to affect teams' performance (Kraut & Streeter, 1995; Van De Ven, Delbecq, & Koenig, 1976). Administrative coordination refers to formal or pre-specified mechanisms used to assign tasks, allocate physical and economic resources, manage resource dependencies, and integrate outputs (Faraj & Sproull, 2000). These mechanisms include budgets, staffing tables, critical path analysis, product road-mapping, milestones, inspections, and review meetings.

Henderson and Clark (1990) observed that a dominant product design usually comes into the market first then the evolving product features are presented by the company after many trials and experiments. Organizations build knowledge and capability around the recurrent tasks that they perform. Since organizations build the routines and practices around core product knowledge, the organizational knowledge and lessons are embedded in the daily administrative coordination practices. The communication channels are built around the major product architecture. Then new changes and releases will be delivered as a small change on the architectural platform.

Both business knowledge and technological knowledge are important for product development. Understanding requirement uncertainty is more about understanding the future business needs and the overall business processes and organizational architecture. Technological knowledge is more involved in the IT infrastructure knowledge and platforms and specific technical knowledge. The ability to leverage the embedded knowledge assets is a key factor for organizations to develop a product successfully. The information processing and knowledge transfer in the product development program will be critical for the final product development performance.

The programs buffer the negative impact of requirement and technological uncertainty on product development. The individual project performances are affected through administrative and expertise coordination between project teams. The inter-project coordination might have a complicated impact on the individual project performances. When the program management team has to prioritize the project needs, some project teams might get the resources but some other teams do not get them. The cost of inter-project coordination might be high for a project team with limited resources.

Based upon Information Processing Theory (IPT) (Galbraith, 1973), a contingency approach is adopted in this study to examine the coordination effects between multiple project teams on the individual project performances in a software product development program (see Figure 7).

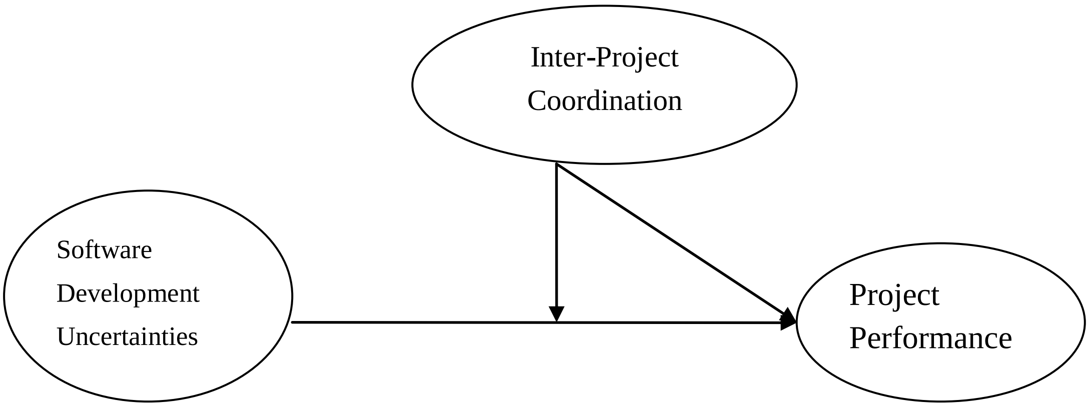

Inter-Project Coordination
Software Development Uncertainties
Project Performance

Figure 7: Theoretical Framework of Inter-project Coordination

Facing critical business and technology changes, this study argues that these changes can be handled by using coordination efforts at a multi-project level, particularly the inter-team administrative coordination and expertise coordination. The key point for this framework is that different types of uncertainty demand various focuses on the efforts and contents of the coordination. The interactions between uncertainty and different types of coordination will have a complicated impact on individual project performances in the program.

3.4 Hypotheses Development

Requirement uncertainty reflects the changes in the business environment and the interests of different stakeholders. Business changes often lead to change in user requirements. The change of user requirements can have an impact on project scope and the project plan at various levels. “Scope creep” will be a typical result because of the changes in user requirements. However the software development program can buffer the impact from requirement uncertainty to the individual projects because software development tasks are distributed among the projects and more resources can be pooled for the changes in a program than in a single project.

However when the users differ among themselves about requirements, it will be a challenging task for a single project team to make difficult choices to decide which and to what extent user requirements should be met. Software development programs can represent multiple projects and work with business users for the diversified requests. Program managers will prioritize the changes of requirements based upon the goals of the product development program and assign the tasks to each individual project team accordingly.

When the requirement uncertainty is high, the formal and informal meetings among multiple project managers and product managers are conducted extensively. Rich information will be exchanged and common understanding of the priority of the requests will be developed. With the supports from the program management team, problems and issues associated with the requirement changes within a single project will be solved in a timely manner and lead to the success of individual project performance.

When requirement uncertainty is low, product features will be developed based upon the original design. Individual project teams can stick to the original plan and develop the products on time and under budget. When requirement uncertainty is high, multiple project teams coordinate collectively in incorporating the requirement changes in the ongoing product development; the product will be responsive to a large range of requirement changes. The individual team’s performance might be affected and evaluated accordingly. Naturally, the following hypothesis is developed.

H1: When requirement uncertainty is high, administrative coordination can moderate the effects of requirement uncertainty on project performance.

When requirement uncertainty is high, the estimation of the impact of the possible change is critical for the product development program. The evaluation of the possible business changes needs business knowledge, component knowledge and architectural knowledge. Sometimes user requirement is not only instable but also ambiguous.

Interpretations are needed and common understandings have to be developed. At this time, expertise can be developed because of distributed, heedful, and emergent processes in the program.

When requirement uncertainty is high and expertise coordination is conducted extensively, close interaction among project teams and project managers, shared common background on the product, and past experience can facilitate the interpretation process. Consequently a common understanding can be built for the product design. The product development program can find the solutions to respond to the changes trhough the inter-project expertise coordination. The solutions will be responsive to changes in the business and lead to a high level of product responsiveness. An individual project team can get a clear task assignment and is more likely to deliver a successful component. Based upon the previous arguments, the following hypothesis is proposed:

H2: When requirement uncertainty is high, expertise coordination can moderate the effects of requirement uncertainty on project performance.

Technologies are understood as the bodies of knowledge, or understanding and practice, that underpin product design and manufacturing (Brusoni et al., 2001). The structure of a product development program is usually built around the current technology and product features. When technology uncertainty is low, the current program structure and the interfaces between multiple project teams are sufficient to handle communication needs.

However when technology uncertainty is high, the current structure of a product development program and the assignment of tasks become inadequate. The current skill sets in the individual project teams are built based upon the past technology needs. When the nature of the tasks is changing, the software product program needs to restructure the procedures and practices of software product development. Formal meetings and informal communications can handle the large amount of information exchange. Therefore when technological uncertainty is high and administrative coordination is performed extensively, the software development program can develop a solution for the technological changes responsively and restructure the project assignments which can lead to individual project success. The above discussion can develop the following hypothesis:

H3: When technology uncertainty is high, administrative coordination can moderate the effects of technology uncertainty on project performance.

Brusoni et al. (2001) argue that usually technological changes are more than the component change and involve the changes in product architecture. Knowledge integration and application on the new problems will be critical for the product development success. Brusoni and Prencipe (2001) examine the role of systems integrator and argue that system integrators act as knowledge and organizational coordinators to guarantee the overall consistency of the product and to orchestrate the network of projects involved in the various stages of design and development. Although system integrators are critical for the product development, the limited number of system integrators and limited time of each integrator require the product development teams to not only knowing where the system integrator is but also look for knowledge and procedures that are stored in the organization such as documents and cases. The team members can interact with the system integrators or gain rich insights from the organizational documented knowledge.

When technology uncertainty is high and when the expertise coordination is high, the distributed expertise in the multiple project teams is pulled together. The interactions between experts and teams are enabled to develop the solutions for the new problems that arise in the product development process because of technological change. The solutions will clarify the technological requirements for each individual project. Individual project performance can be achieved with clear technological requirements and goals. The integration of different product components will also be responsive to the technological changes. Therefore it is proposed that

> *H4: When technology uncertainty is high, expertise coordination can moderate the effects of technology uncertainty on project performance.*

3.5 Research Methodology

A survey was used to collect data and test the hypotheses. Table 21 in 3.9 Appendix lists all the specific survey items.

3.5.1 Measures

All the items were measured on a five-point Likert scale, ranging from “to a large extent” (5) to “not at all” (1). The constructs were:

a. *Inter-project Administrative Coordination*: The measure for inter-project administrative coordination had six items from Kraut and Streeter (1995). It included both the formal and interpersonal administrative coordination practices. A sample item was “the extent of using formal policies and procedures for coordinating the projects in the program”.

b. Inter-project Expertise Coordination: The measure for inter-project expertise coordination was adapted from Faraj and Sproull (2000) with four items for knowing expertise location, three items for recognizing where expertise is needed and four items for bringing expertise to bear.

c. Requirement uncertainty: The measure for requirement uncertainty was adapted from Nidomolu (1995). It had three dimensions. Requirement instability is described by the extent of change in user requirements over the course of product development and had three items. Requirement diversity is described by the extent to which users differed amongst themselves in their requirements and had three items. Requirement analyzability refers to the extent to which a conversion process can be reduced to mechanical steps or objective procedures and had four items.

*d. Technology uncertaint*y: The measure for technology uncertainty was adapted from Nidomolu (1995). It had two dimensions. Technological unpredictability describes the extent to which unexpected and novel technology occur during the software development product process and had four items. Technological analyzability describes the extent to which the task of converting requirement specifications to software could be undertaken using well-established procedures. Technological analyzability had eight items.

e. *Project performance:* Project performance must represent many aspects of the development process and has been recognized as an important construct by the past literature. The measure of project performance included seven items (ability to meet project goals, expected amount of work completed, quality of work completed, adherence to schedule, adherence to budget, efficient task operations and high work morale) and required the respondents to answer based on the most recently completed projects in the program (1 – Never, 5 – Always) (Nidumolu, 1995).

3.5.2 Data Collection

Data was collection in the Mainland in 2007-2008. The data collection unit was a “program”. On average each program included 3-5 individual IT projects. For each program, a project manager is identified and invited to fill in the questions about the inter-project coordination within the program. The recruiting method for participants was snowballing. Investigators’ friends who worked in IT software companies were invited to participate in the survey and asked to introduce more participants. The current number of valid responses is 70. The small sample size can cause the issue of statistical power (Marcoulides & Saunders, 2006). Another concern is common method bias. This issue has been addressed by testing Harman’s one factor analysis (Podsakoff, MacKenzie, Lee, & Podsakoff, 2003) and the first factor extracted variance less than 30%.

Table 13: List of construct in Study II

Construct	**Items**	**Loadings**	**T-Statistic**	**Item- total**	**Conbach Alpha**	**Composite Reliability**	**Variance extracted**
Administrative Coordination	AC1	0.81	13	0.63	0.77	0.85	0.59
	AC2	0.82	17.55	0.64			
	AC3	0.7	8.64	0.48			
Knowing where expertise is	ECL1	0.73	5.75	0.55	0.84	0.89	0.68
	ECL2	0.86	29.35	0.72			
	ECL3	0.82	15.21	0.66			
	ECL4	0.87	30.58	0.75			
Knowing when expertise is needed	ECN1	0.87	29.19	0.80	0.81	0.92	0.79
	ECN2	0.93	38.22	0.62			
	ECN3	0.87	19.85	0.62			
Bringing expertise to bear	ECB1	0.72	7.89	0.53	0.84	0.87	0.63
	ECB2	0.85	20.32	0.86			
	ECB3	0.82	14.17	0.64			
	ECB4	0.78	16.19	0.87			
Requirement instability	RS1	0.63	7.13	0.41	0.75	0.84	0.58
	RS2	0.85	25.25	0.67			
	RS3	0.86	27.55	0.68			
	RS4	0.66	6.74	0.43			
Requirement diversity	RD1	0.88	34.74	0.71	0.82	0.89	0.73
	RD2	0.83	20.41	0.63			
	RD3	0.85	22.53	0.66			
Requirement unanalyzability	RA1	0.86	18.16	0.6	0.75	0.89	0.74
	RA2	0.86	21.03	0.6			

Table 13: List of construct in Study II (Continued)

Construct	**Items**	**Loadings**	**T-Statistic**	**Item- total**	**Conbach Alpha**	**Composite Reliability**	**Variance extracted**
Technology unanalyzability	TA1	0.88	33.51	0.78	0.9	0.93	0.77
	TA2	0.86	24.46	0.75			
	TA3	0.91	45.59	0.84			
	TA4	0.84	25.86	0.73			
Technology unpredictability	TP1	0.76	11.73	0.67	0.9	0.92	0.6
	TP2	0.73	10.96	0.64			
	TP3	0.83	15.93	0.76			
	TP4	0.84	23.57	0.78			
	TP5	0.83	19.46	0.78			
	TP6	0.68	9.13	0.78			
	TP7	0.77	14.78	0.78			
	TP8	0.72	7.21	0.78			
Project performance	PP1	0.81	11.18	0.65	0.86	0.89	0.54
	PP2	0.86	22.18	0.77			
	PP3	0.73	14.57	0.70			
	PP4	0.70	12.74	0.68			

3.6 Data Analysis

3.6.1 Demographic Information

Demographic information of respondents is shown in Table 14. Of those participants who provide gender information, 84.3% were male and 12.9% were female. The respondents consisted of 11 project leaders and 33 project managers and other managers involved in the product development process. The average year of experience in the IT industry was 7.62 year and the average year of experience in the current company was 4.37 year. Each respondent had 12 subordinates on average. Among the respondents, 58.6% worked in the IT industry. About 42.9% of the companies had employees less than 500 but more than 50. About 18.6% of the companies had more than 1000 employees. According to the respondents, 48.6% of the projects were completed less than 1 year and 34.3% were completed in less than 2 years. Among the respondents, 12.9% received PMP certification and 25.7% were currently pursuing the certification. Overall, the pool of respondents was well qualified to judge the issues related to inter-project coordination and project performance.

3.6.2 PLS Analysis

The hypotheses were tested and verified by using partial least square (PLS) analysis (Lohmoller, 1989). This is a latent structural equation modeling technique that uses a component-based approach to estimation; it contains two steps. The first examines the measurement model and the second assesses the structural model. When using PLS, researchers must pay attention to three concerns: (1) the reliability and validity of measures;

(2) the appropriate nature of the relationship between measures and constructs; and (3) path coefficient, model adequacy, and the final model from the available set of alternatives (Hulland, 1999). PLS-Graph Version 3.00 was used to test the hypotheses.

3.6.3 Measurement Model

Item reliability, convergent validity, and discriminant validity tests were examined for the construct measures. Individual item reliability can be examined by observing the factor loading of each item. High loading implies that the shared variance between the construct and its measurement is higher than the error variance. Factor loading higher than 0.7 can be viewed as highly reliable and factor loadings less than 0.5 should be dropped.

Convergent validity is assured when multiple indicators are used to measure one construct. It can be examined by insistence on high reliability of the questions, composite reliability of the constructs, and consideration of the variance extracted by constructs (AVE). AVE, proposed by Fornell and Larcker (1981), considers the variance captured by the indicators. If the AVE is less than 0.5, the variance captured by the construct is less than the measurement error and the validity of the single indicator and construct is questionable. Table 13 shows that the variance extracted for each construct was larger than 0.5, and the item-construct correlations were all be more than 0.7.

Construct reliability is demonstrated by Cronbach’s alpha. Composite reliability of constructs is calculated by squaring the sum of the loadings then dividing it by the sum of the squared loadings plus the sum of the error terms (Werts et al., 1974). Table 13 shows that

Table 14: Demographic data of project managers in Study II

Variables	Categories	#	%
Gender	Male	59	84.3
	Female	9	12.9
	Missing	2	2.9
	Project Member	2	22.9
Position	System Analyst	2	2.9
	Project Leader	11	15.7
	Project Manager	33	47.1
	Program Manager	9	12.9
	Customers	0	0
	Product Director	1	1.4
	Product Manager	6	8.6
	Missing	6	8.6
Industry	IT industry	41	58.6
	Non-IT industry	26	37.1
	Missing	3	4.3
Company size	<=50	19	27.1
	50-500	30	42.9
	500-1000	6	8.6
	>1000	13	18.6
	Missing	2	2.9
Average project duration	< 1 year	34	48.6
	1-2 years	24	34.3
	2-3 years	6	8.6
	3-5 years	3	4.3
	>=6 years	1	1.4
	Missing	2	2.9
PMP certification	Certified	9	12.9
	Pursuing	18	25.7
	Intend to pursue	15	21.4
	Not certified	20	28.6
	Missing	8	11.4
Work experience		7.62	years
Current company experience		4.37	years
Number of subordinates		12	
Total Sample size		70	

composite reliability of each construct is also above 0.7 which was acceptable.

Discriminant validity determines whether the measures of the constructs are different from each other (Messick, 1980). It can be assessed by testing whether the square root of AVE is larger than correlation coefficients. Loading values for each indicator (shown in bold font in Table 15) exceeded 0.7, which indicated high and significant discriminant validity. To have discriminant validity, indicators should have higher loadings in the corresponding constructs than in other constructs. Procedures suggested by Smith, Keil & Depledge (2001) were used to generate cross-loading values. Loading values for each indicator (shown in bold font in Table 15) exceeded 0.7, which indicated high and significant discriminant validity.

Basic information about each variable is given in Table 16 including means, standard deviation, skewness and kurtosis. For each variable the skewness was less than 2 except complexity-in-use dimension and the kurtosis less than 5, indicating no significant violation of normal distribution (Ghiselli et al., 1981). The construct score of each variable obtained from PLS was used to calculate the hierarchical moderation effect in SPSS.

The hierarchical moderation test is used to analyze the moderating effect. Following the suggestion from Carte and Russell (Carte & Russell, 2003), moderating effect can be assured by comparing the difference between main effect model and moderating effect model. This hierarchical process was adopted by many IS researchers (Gefen et al., 2000; Khalifa & Cheng, 2002; Limayem et al., 2001; Mathieson et al., 2001). A four-step analysis process was used.

We first obtained the R-square (R_1^2) of the main effect model which includes independent variable (IV, i.e. uncertainty), moderator (i.e., coordination), and dependent variable (DV, i.e., project performance) only. Then the R-square (R_2^2) of the moderating effect model was obtained by including IV, moderator, interaction term (i.e., the interaction of uncertainty and coordination), and dependent variable in the model. The interaction term used in the model is calculated by adding the multiplying result between each indicator in independent variable and each indicator in moderator (Chin, 2003). Third, we derived an estimated effect size of f^2 from (1- R_2^2) / (R_2^2 - R_1^2) and then obtained pseudo F-value by multiplying f^2 with $(n-k-1)$ where n is the sample size and k is the number of independent variable in the regression equation. f^2 score of 0.03, 0.15, and 0.35 imply small, moderate, and large interaction effects (Cohen, 1988). Finally, we compare the pseudo F-value with F1, n-k-1. The above four steps can test the change of variance extracted by adding a new variable (the interaction term) into the model.

Table 15: Cross-factor loadings in Study II

	Administrative Coordination	Knowing where expertise is located	Knowing when expertise is needed	Bringing expertise to bear	Project Performance	Requirement Instability	Requirement diversity	Requirement unanalyzability	Technology Analayzability	Technology Unpredictability
AC1	**0.79**	0.39	-0.04	-0.01	0.51	0.07	-0.07	-0.32	-0.36	-0.12
AC2	**0.68**	0.41	0.03	-0.07	0.32	0.13	0.14	-0.36	-0.39	-0.08
AC3	**0.81**	0.50	-0.15	0.00	0.34	0.30	0.31	-0.34	-0.18	0.01
ECL1	0.63	**0.73**	-0.23	-0.06	0.32	0.15	0.16	-0.29	-0.27	-0.03
ECL2	0.50	**0.86**	-0.18	-0.02	0.19	0.21	0.16	-0.17	-0.23	-0.01
ECL3	0.35	**0.82**	-0.25	-0.06	0.13	0.07	0.01	-0.02	-0.15	0.04
ECL4	0.45	**0.87**	-0.27	-0.23	0.19	0.14	0.21	-0.25	-0.40	0.03
ECN1	-0.12	-0.28	**1.00**	-0.01	-0.06	-0.06	-0.01	0.24	0.16	-0.09
ECN2	-0.21	-0.17	**0.70**	-0.02	-0.17	-0.08	-0.15	0.26	-0.02	-0.14
ECN3	0.06	-0.08	**0.69**	-0.01	0.13	0.13	0.10	0.00	-0.03	-0.09
ECB1	0.10	-0.06	-0.02	**0.56**	0.07	0.16	0.00	-0.16	0.08	-0.10
ECB2	-0.04	-0.13	-0.01	**1.00**	0.06	0.08	0.08	-0.07	0.11	-0.11
ECB3	0.13	0.03	-0.02	**0.71**	0.21	0.22	0.17	-0.21	-0.06	-0.12
ECB4	-0.04	-0.13	0.00	**1.00**	0.06	0.08	0.09	-0.07	0.12	-0.12
RS1	0.27	0.20	0.04	0.03	0.34	**0.63**	0.35	-0.35	-0.28	-0.07
RS2	0.21	0.24	-0.17	0.08	-0.04	**0.85**	0.56	-0.28	-0.09	0.24
RS3	0.31	0.13	-0.15	0.08	0.23	**0.86**	0.57	-0.34	-0.25	0.13
RS4	-0.05	-0.06	0.20	0.05	-0.14	**0.66**	0.56	-0.12	0.02	0.24

(Table 15 Continued)

	Administrative Coordination	Knowing where expertise is located	Knowing when expertise is needed	Bringing expertise to bear	Project Performance	Requirement Instability	Requirement diversity	Requirement unanalyzability	Technology Analyzability	Technology Unpredictability
RD1	0.15	0.10	-0.03	0.07	0.21	0.64	**0.88**	-0.38	-0.23	0.24
RD2	0.31	0.30	0.03	0.06	0.27	0.53	**0.83**	-0.55	-0.21	0.23
RD3	-0.03	0.04	-0.04	0.08	0.01	0.56	**0.85**	-0.31	-0.15	0.39
RA1	-0.44	-0.23	0.40	-0.05	-0.42	-0.32	-0.44	**0.86**	0.38	-0.24
RA2	-0.33	-0.09	0.11	-0.06	-0.29	-0.23	-0.38	**0.86**	0.28	-0.18
TA1	-0.36	-0.35	0.14	0.12	-0.37	-0.23	-0.20	0.35	**0.88**	0.01
TA2	-0.40	-0.35	0.27	0.11	-0.38	-0.17	-0.17	0.39	**0.87**	0.06
TA3	-0.26	-0.26	0.08	0.07	-0.41	-0.12	-0.23	0.32	**0.91**	-0.04
TA4	-0.18	-0.18	0.06	0.08	-0.29	-0.19	-0.21	0.29	**0.84**	-0.02
TP1	-0.03	-0.04	-0.05	-0.06	-0.01	0.08	0.21	-0.19	0.01	**0.76**
TP2	-0.01	-0.02	0.05	-0.08	-0.11	0.22	0.26	-0.05	0.00	**0.73**
TP3	0.03	0.05	-0.04	-0.05	0.00	0.23	0.36	-0.24	0.02	**0.83**
TP4	-0.09	0.06	-0.08	-0.09	-0.19	0.07	0.25	-0.13	0.07	**0.84**
TP5	-0.14	-0.07	-0.08	-0.10	-0.09	0.06	0.14	-0.19	0.12	**0.83**
TP6	-0.04	-0.03	-0.14	-0.15	-0.11	0.09	0.28	-0.25	-0.07	**0.68**
TP7	-0.10	0.05	-0.11	-0.13	-0.01	0.28	0.32	-0.30	-0.17	**0.77**
TP8	-0.04	0.04	-0.02	-0.04	-0.21	0.14	0.25	-0.13	0.01	**0.72**
PP1	0.38	0.32	0.03	0.05	**0.79**	0.10	0.11	-0.30	-0.52	-0.20
PP2	0.42	0.24	-0.13	0.07	**0.88**	0.18	0.23	-0.43	-0.38	-0.05
PP3	0.42	0.18	-0.14	0.04	**0.84**	0.09	0.23	-0.41	-0.31	-0.04
PP4	0.26	0.09	0.05	0.06	**0.83**	0.04	0.06	-0.32	-0.18	-0.12

Table 16: Descriptive Statistics in Study II

	Mean	**S.D.**	**Skewness**	**Kurtosis**	**ac**	**ecl**	**ecn**	**ecb**	**rs**	**rd**	**ra**	**ta**	**tp**	**pp**
Administrative Coordination (ac)	3.48	0.83	-0.63	.12	**0.77**									
Knowing the expertise Location (ecl)	3.57	0.87	-1.15	1.45	0.58	**0.82**								
Knowing when expertise is needed (ecn)	2.71	1.01	0.32	-0.53	0.21	-0.06	**0.89**							
Bringing the expertise to bear (ecb)	3.28	0.91	-0.56	0.51	0.66	0.53	0.13	**0.79**						
Requirement instability (rs)	3.44	0.93	-0.01	-0.33	0.23	0.17	0.18	0.15	**0.76**					
Requirement Diversity (rd)	3.41	0.93	-0.33	-0.48	0.18	0.17	-0.07	0.13	0.68	**0.85**				
Requirement unanalyzability (ra)	1.33	0.79	0.98	1.75	-0.47	-0.22	-0.14	-0.26	-0.36	-0.48	**0.86**			
Technological unanalyzability (ta)	1.70	1.19	1.02	1.31	-0.43	-0.32	-0.20	-0.28	-0.20	-0.23	0.39	**0.88**		
Technological unpredictability (tp)	2.63	0.75	0.21	-0.25	-0.10	0.01	0.07	-0.16	0.19	0.34	-0.24	0.00	**0.77**	
Project performance (pp)	3.81	0.58	-0.31	0.86	0.49	0.25	0.15	0.36	0.12	0.19	-0.44	-0.41	-0.12	**0.73**

Table 17: Overall results for the hypotheses in Study II

R-square change		Administrative Coordination	Inter-project Expertise Coordination		
			Knowing the location of expertise	Knowing when expertise is need	Bringing expertise to bear
Requirement Uncertainty	Requirement instability	0.004	0.016	0.013	.002
	Requirement diversity	0.058**	0.001	0.048 **	0.034
	Requirement Unanalyzability	0.005	0.006	0	0.001
Technological Uncertainty	Technology unanalyzability	0.007	0.017	0.018	0
	Technology unpredictability	0.046**	0.001	0.002	0.002

Table 17 shows the hierarchical moderation results for the interaction between uncertainty and inter-project coordination. Hypothesis 1 was partially supported since only the moderating effect of administrative coordination on requirement diversity was supported. Hypothesis 2 states the moderating effect of expertise coordination on requirement uncertainty and got partial support. The interaction effect of knowing when expertise is needed on requirement diversity is significant. The moderating effect of bringing expertise to bear on requirement diversity was supported with the significant R square change.

Hypothesis 3 on the interaction of administrative coordination and technological uncertainty was partially supported because of the significant change of effect size after adding the interaction term of administrative coordination and technological unpredictability. Hypothesis 4 which states the moderating effect of expertise coordination on technology uncertainty was not supported by the data analysis results.

To better illustrate the moderating effects, I graphed the interaction effects following the procedures set forth by Cohen and Cohen (1987). Figure 8 shows the moderating effect of administrative coordination on the relationship between requirement diversity and project performance (See Table 20 in 3.9 Appendix). While R square change in project performance was significant, it is not enough to simply assume that the interaction graph demonstrates that the change in performance is significantly different than zero without testing for the significance of the slope (Aiken & West, 1991). The path coefficient of the interaction of administrative coordination and requirement diversity, -0.266, was significant. Specifically, the slope significance test demonstrated that when requirement diversity is low, the high level of administrative coordination has a higher impact on project performance than the low level of administrative coordination. In contrast, when requirement diversity is high, the low level of administrative coordination will more likely to lead to better project performance than the high level of administrative coordination.

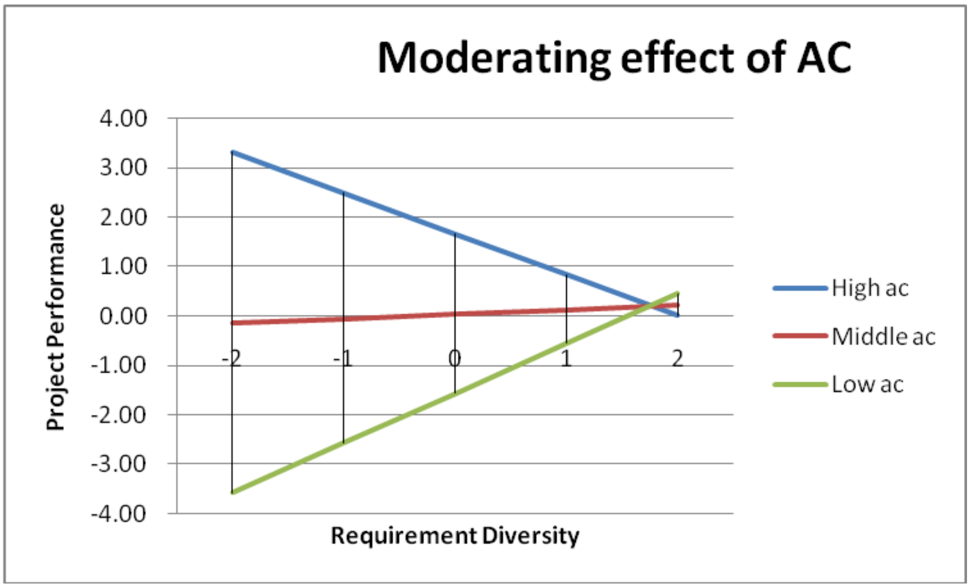


Figure 8: The moderating effect of administrative coordination on requirement diversity

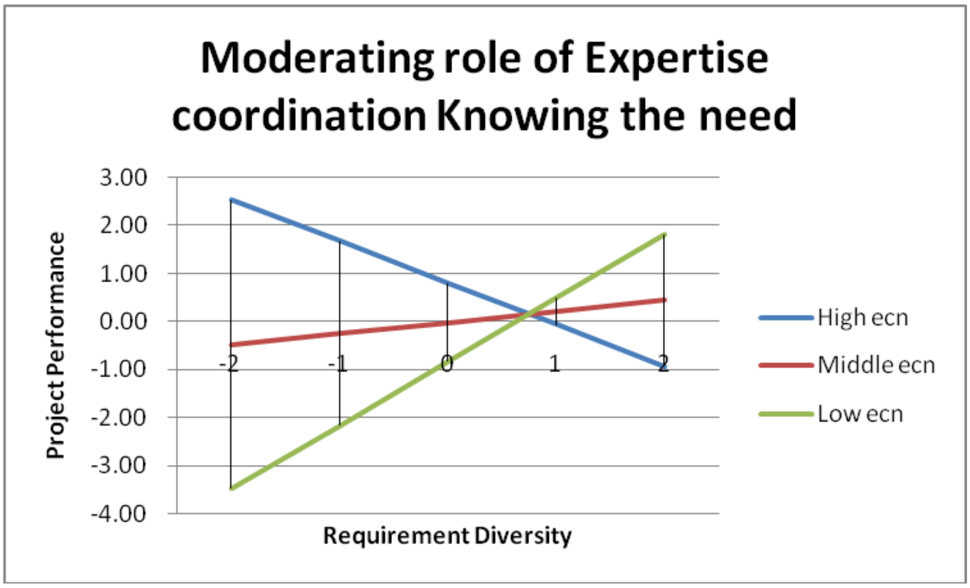


Figure 9: The moderating effect of knowing when expertise is needed on requirement diversity

Figure 9 shows the moderating effect of knowing when expertise is needed on the relationship between requirement diversity and project performance (See Table 19 in 3.9 Appendix). While R square change in project performance was significant, it is not enough to simply assume that the interaction graph demonstrates that the change in performance is significantly different than zero without testing for the significance of the slope (Aiken & West, 1991). The path coefficient of the interaction of knowing when expertise is needed and requirement diversity, -0.282, was significant. Specifically, the slope significance test demonstrated that when requirement diversity is low, the high level of “knowing when expertise is needed” has a higher impact on project performance than the low level of “knowing when expertise is needed”. In contrast, when requirement diversity is high, the low level of “knowing when expertise is needed” is more likely to lead to better project performance.

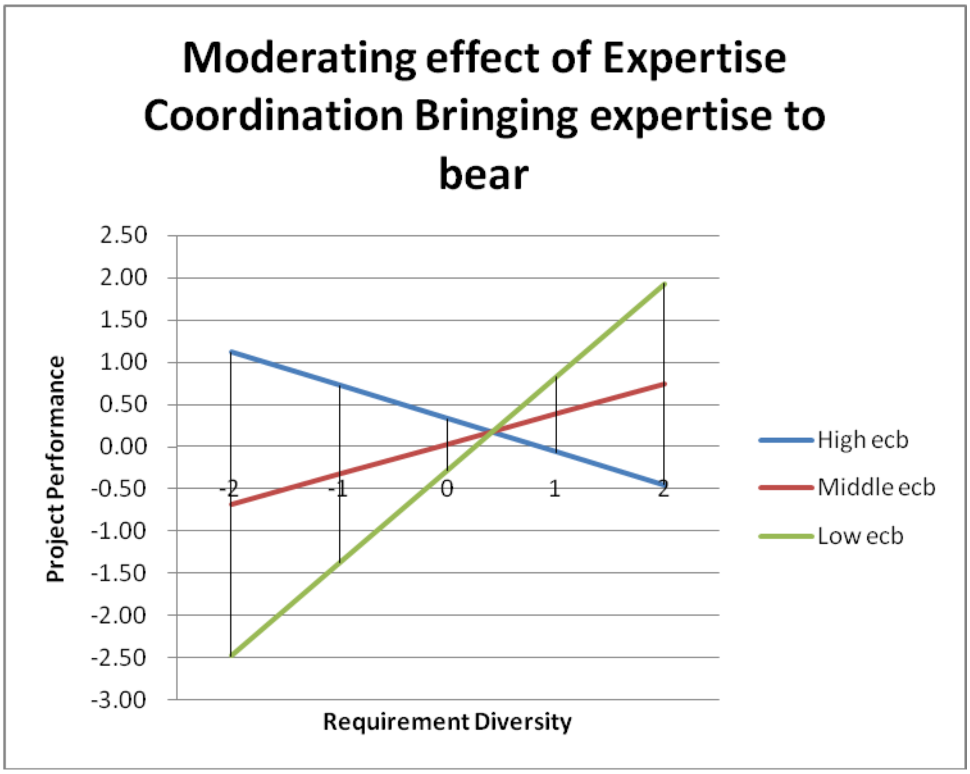


Figure 10: The moderating effect of bringing expertise to bear on requirement diversity

Figure 10 shows the moderating effect of bringing the expertise to bear is needed on the relationship between requirement diversity and project performance (See Table 19 in 3.9

Appendix). The path coefficient of the interaction of bringing the expertise to bear and requirement diversity, -.213, was significant. It demonstrates that the change in performance is significantly different than zero without testing for the significance of the slope (Aiken & West, 1991). Specifically, the slope significance test demonstrated that when requirement diversity is low, the high level of "bringing the expertise to bear" has a higher impact on project performance than the low level of "bringing the expertise to bear". In contrast, when requirement diversity is high, the low level of "bringing the expertise to bear" is more likely to lead to better project performance.

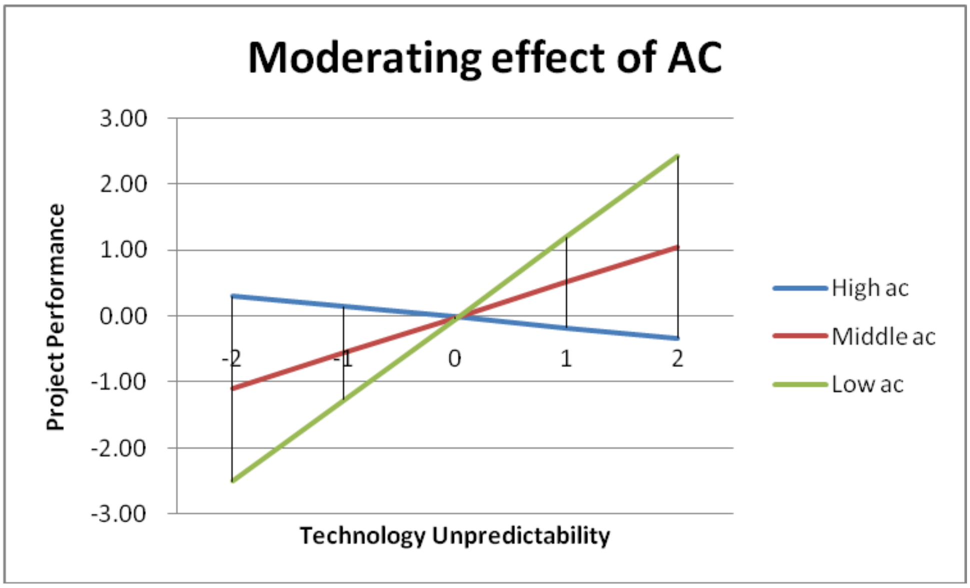


Figure 11: The moderating effect of administrative coordination on technology unpredictability

Figure 11 shows the moderating effect of administrative coordination is needed on the relationship between technological unpredictability and project performance (See Table 18 in 3.9 Appendix). The path coefficient of the interaction of administrative coordination and technological unpredictability, -.263, was significant. It demonstrates that the change in performance is significantly different than zero without testing for the significance of the slope (Aiken & West, 1991). Specifically, the slope significance test demonstrated that when technological unpredictability is low, the high level of administrative coordination has a higher impact on project performance than the low level of administrative coordination. In contrast, when technological unpredictability is high, the low level of administrative coordination is more likely to lead to better project performance.

3.7 Discussion

The purpose of this study is to answer the research question *"How can a software development program manage the software product development uncertainties through inter-project coordination? "* The inter-project coordination in the programs has a mixed impact on individual project performances. Generally speaking, administrative coordination has a significant role in improving the individual project performance. Inter-project administrative coordination moderates the relationship between requirement diversity and individual project performance. The product development program leverages the program's advantages by using administrative coordination to distribute the impacts of requirement changes among multiple projects and make a coherent plan for the changes within a complex network of projects. Project structuring and support is most important in the development phase of the project (Hoegl & Weinkauf, 2005). So product planning and structuring play critical role in the software product development. When the product plan is well executed, individual projects have clear goals and performance evaluation criteria. Therefore, when requirement diversity is low, projects are structured based upon the original plan and a high level of administrative coordination can lead to higher project performance than a low level of administrative coordination with a poor product plan. When requirement diversity is high, the projects are usually structured based upon different functions and do not share many common backgrounds. The low level of administrative coordination will enable the project team to concentrate on its own problems and get less distracted by the inter-project communication. Therefore when requirement diversity is high, the low level of administrative coordination has more positive impacts on project performance than the high level of administrative coordination.

The data results also supported the moderating effect of administrative coordination on technological unpredictability. When the technology used in product development remains current, the project structuring and task assignments can be kept same as the original plan. A high level of administrative coordination will provide sufficient communication between project teams and lead to a high level of project performance. Brusoni et al. (2001) argue that multi-technology firms need to have knowledge in excess of what they need for what they make, to cope with imbalances caused by uneven rates of development in the technologies on which they rely and with unpredictable product-level interdependencies. When technology develops quickly and new technology has to be applied in product development, projects have to be restructured and tasks have to be re-designed. These changes have fundamental changes on individual project performances. Project escalation and closure cannot be avoided.

Therefore when technological unpredictability is high, a low level of administrative coordination will not change the project goals and more likely lead to individual project success than a high level of administrative coordination.

Expertise coordination partially moderates the relationship between requirement diversity and project performance. When requirements are diversified, component compatibility and product integration will become crucial for product success. If the project teams are able to recognize the need for expertise early, integration issues can be solved quickly and the project success likelihood will be high. However as the level of requirement diversity is increasing, the project teams can easily recognize the compatibility problems. The project teams will benefit from the constant attention to working with experts on the integration issues. A high level of effects in recognizing when the expertise is needed becomes unnecessary and even has reversed impacts on project performance.

Feed-forward learning flow and feedback learning flow are embedded in the inter-project expertise coordination within a software product development program. The software development program provides strong supports to the learning flow by enabling the interactions between expert individuals and individual project teams who are usually concealed by the project boundary. The feed-forward flow may begin with individuals' intuitive insights and experiences. When requirements are diversified, project team members develop the insights and experiences on the development of particular components. Recognizing when expertise is needed and bringing the expertise to the project teams will provide the experts and individual project teams an opportunity of sharing the perceptions for new problems and issues. Consequently, shared understandings emerge and becomeintegrated into a sense of collective actions.

Feedback learning relates to the way in which institutionalized learning (culture, structures, systems, procedures, and strategy) affects individuals and groups. Program management team also focuses on the institutionalization of learning by documenting the changes, updated the roadmaps of products, formalizing some procedures and sharing the experiences across projects. The learning is reinforced and improves the efficiency and effectiveness of future product development.

3.7.1 Implications

This study provides in-depth understandings of inter-project coordination within a program and its effects on individual project performances. The program management team plays a critical role in project structuring and support. However, the product plan has to be adaptive to requirement uncertainty. Good efforts of risk estimation and planning can save the program team a lot of time and resources in adjusting the project goals and tasks to respond to the changes.

However, technological uncertainty has fundamental influences on the project structuring and performance evaluation. The program management team should try to avoid the changes on the existing and ongoing projects and adapt to the change by initiating new projects and balancing the risks in different project portfolio.

Another implication is about learning and expertise coordination. Although system integrators play a critical role in product integration, many times component changes initiate the essential changes in the product. Project team members have the first hand experiences and intuition that should receive attention from the program management team. Well established procedures of project learning and product development can prepare the program well for future changes and solution search processes.

3.7.2 Limitations and future research

As other studies, this study has several limitations. This study only examines a set of moderators. Many moderators that can affect the relationship between uncertainty and project performance not examined. The second limitation is the sample size. The statistical power is limited because of small sample size. The third limitation is that the interpretation of the results is limited by culture factor. Future research can test the validity of the conclusions by using multiple sources of data or a combination of quantitative and qualitative methods.

The program management team plays a critical role in building the program's procedures and routines of coordination. A longitude study of the development process of these routines can deepen the understanding of inter-project coordination within a program. In addition, the communication style and attitudes for risks of project managers and program managers can influence the effectiveness of coordination. It will be interesting to explore the risk aversion behaviors of project managers and program managers and its impact on the project team performances and the overall product development.

3.8 Conclusions

What differentiated this study from previous efforts was that this study focuses on the inter-project coordination issues in a software product development program. Although planning and project structuring are critical for the product development, exceptions and unexpected events have to be handled effectively and efficiently for the final product delivery. Project managers should focus on the inter-project team coordination and identify the impacts of changes on their own project performances and the integration of project deliveries for the final product. The ongoing exchange of information and communication will give more room for the emergent planning and problem-solving across the project boundaries.

Organizations should spend more time and efforts in managing the expertise resources in the software development program. Experts, system integrators and the persons who have the overall architectural knowledge should be involved in the decision makings to solve the new problems created by the changes and identify the implicated changes in the overall architecture. According to the collective mind theory (Weick & Roberts, 1993), when the environment is difficult to sense and interpret, organizations need to develop “mindfulness”. Weick and Roberts (Weick & Roberts, 1993) have argued for the deference to expertise. During troubled times, shift the leadership role to the person or team possessing the greatest expertise and experience to deal with the problem at hand. Provide them with the empowerment they need to take timely, effective action. Avoid using rank and status as the sole basis for determining who makes decisions when unexpected events occur.

3.9 Appendix of Chapter Three

Table 18: Hierarchical moderation results of inter-project coordination on technological uncertainty

Path Coefficient (Standard error)	Model 0	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7	Model 8
Intercept	-.011	-.047	-.052	.038	-.012	-.023	-.009	-.015	-.019
Technology unanalyzability (TA)	-.248**	-.272**	-.258**	-.246**	-.251**	-.243**	.-.251	-.246	-.249
Technology unpredictability (TP)	-.057	-.061	.022	-.04	-.055	-.075	-.044	-.057	-.047
Administrative Coordination (AC)	.430**	.441**	.453**	.415**	.434**	.398**	.449	.423	.446
Expertise Location (EL)	-.065	-.054	-.035	-.123	-.067	-.036	-.073	-.065	-.049
Expertise Need (EN)	.000	.000	.003	.003	.000	.101	-.027	.000	.000
Expertise Bring to Bear (EB)	.008	.008	.011	.009	.008	.007	.008	.025	-.024
Interaction		AC*TA	AC*TP	EL*TA	EL*TP	EN*TA	EN*TP	EB*TA	EB*TP
		-.079	-.263**	.173	-.033	-.072	-.039	-.016	-.04
R2	.319	.326	.365	.336	.32	.337	.321	.319	.321
R2 Change		.007	.046**	.017	.001	.018	.002	0	.002

Table 19: Hierarchical moderation results of inter-project expertise coordination on requirement uncertainty

Path Coefficient (Standard error)	Model 0	Model 1	Model 2	Model 3	Model4	Model5	Model6	Model7	Model8	Model9
Intercept	-.011	-.038	-.019	-.026	.004	-.035	-.012	.006	.019	-.001
Requirement Instability (RS)	-.099	-.1	-.099	-.109	-.094	-.047	-.102	-.093	-.08	-.09
Requirement Diversity (RD)	.051	.049	.049	.056	.054	.06	.054	.039	.012	.04
Requirement Unanalyzability (RA)	-.264*	-.269*	-.273*	-.283*	-.288**	-.289**	-.264*	-.273*	-.249*	-.264*
Administrative Coordination (AC)	.411**	.377**	.398**	.398**	.417**	.408**	.409**	.385**	.367**	.394**
Expertise Location (EL)	-.018	.017	-.012	-.016	-.038	-.054	-.016	-.043	-.061	-.012
Expertise Need (EN)	.002	.002	.003	-.001	-.05	-.022	.009	.003	.002	.003
Expertise Bring to Bear (EB)	.005	.007	.006	.006	.006	.007	.005	.068	.133*	.023
Interaction		EL*RS	EL*RD	EL*RA	EN*RS	EN*RD	EN*RA	EB*RS	EB*RD	EB*RA
		.145	.04	-.085	-.098	-.282**	-.003	-.110	-.213*	.044
R2	0.322	.338	.323	.328	.335	.37	.322	.334	.356	.323
R2 Change		.016	.001	.006	.013	.048**	0	.002	.034*	.001

Table 20: Hierarchical moderation results of inter-project administrative coordination on requirement uncertainty

Path Coefficient (Standard error)	Model 0	Model 1	Model 2	Model 3
Intercept	-.011	.003	.039	.016
Requirement Instability (RS)	-.099	-.091	-.079	-.073
Requirement Diversity (RD)	.051	.044	.054	.041
Requirement Unanalyzability (RA)	-.264*	-.273*	-.206	-.243
Administrative Coordination (AC)	.411**	.416**	.458**	.394**
Expertise Location (EL)	-.018	-.016	.005	-.004
Expertise Need (EN)	.002	.003	.002	.003
Expertise Bring to Bear (EB)	.005	.005	.004	.005
Interaction		AC*RS	AC*RD	AC*RA
			-.266**	
R2	0.322	0.326	0.380	0.327
R2 Change		.004	0.058**	0.005

Table 21: Survey items in Study II

Construct	Labels	Items
Administrative Coordination	AC1	the extent of using program documents and memos
	AC2	the extent of regularly scheduled meetings of project managers
	AC3	the extent of informal or unplanned discussions
Knowing where expertise is	ECL1	a good map of expertise in this program
	ECL2	expertise is assigned to projects according to skills
	ECL3	experts know their skills related to projects
	ECL4	project managers know who in the program has skills related to projects
Knowing when expertise is needed	ECN1	some projects lack of certain knowledge to achieve project goals
	ECN2	some project teams do not have necessary knowledge to perform the tasks.
	ECN3	some project teams do not have enough knowledge
Bringing expertise to bear	ECB1	project teams in the program share knowledge

	ECB2	project teams in the program are willing to share knowledge
	ECB3	there are exchange of information and knowledge among project teams.
	ECB4	project teams freely provide special knowledge in the program.
Requirement instability	RS1	requirements fluctuated in early phase of product development
	RS2	requirements fluctuated in later phase of product development
	RS3	requirements identified at the beginning were quite different from those at the end.
	RS4	requirements will fluctuate in the future
Requirement diversity	RD1	clients differ a great deal among themselves in requirements.
	RD2	Efforts have to be spent in requirement reconciliation.
	RD3	It is difficult to satisfy one group of clients without reducing support to others.
Requirement analyzability	RA1	Available knowledge helps in requirement conversion.
	RA2	a sequence of steps can be followed for requirement conversion.
Technology analyzability	TA1	a clear known way to develop the product
	TA2	available knowledge helps in product devleopment.
	TA3	established procedure can be used in product development
	TA4	a sequence of steps can be followed for product development.
Technology predictability	TP1	the extent of predicting the problems in hardware platform used in product development
	TP2	the extent of predicting the problems in software platform in product development
	TP3	the extent of predicting the problems in programming language
	TP4	the extent of predicting the problems in telecommunication technology
	TP5	the extent of predicting the problems in database technology
	TP6	the extent of predicting the problems in design techniques
	TP7	the extent of predicting the problems in in coding and testing
	TP8	the extent of predicting the problems in product installation
Project Performance	PP1	project was completed on schedule
	PP2	project accomplished all the tasks
	PP3	project had efficient task operations
	PP4	project had maintained high morale

3.10 Chapter Three List of References

Aiken, L. S., & West, S. G. (1991). *Multiple Regression: Testing and Interpreting Interactions*. Thousand Oaks, CA.: Sage.

Boehm, B., W. . (1991). Software Risk Management: Principles and Practices. *IEEE Software, 8*(1), 32.

Brown, S. L., & Eisenhardt, K. M. (1995). Product Development: Past Research, Present Findings, and Future Directions. *Academy of Management Review, 20*(2), 343-378.

Brusoni, S., & Prencipe, A. (2001). Unpacking the black box of modularity: Technologies, products and organizations. *Industrial and Corporate Change, 10*(1), 179.

Brusoni, S., Prencipe, A., & Pavitt, K. (2001). Knowledge specialization, organization coupling, and the boundaries of the firm: Why do firms know more than they make? *Administrative Science Quarterly, 46*(4), 597.

Burgelman, R. A., Maidique, M. A., & Wheelwright, S. C. (1996). *Strategic management of technology and innovation*. Chicago: Irwin.

Carte, T. A., & Russell, C. J. (2003). In Pursuit of Moderation: Nine Common Errors and Their Solutions. *MIS Quarterly, 27*(3), 479-501.

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, New Jersey: Lawrence Erlbaum.

Cohen, J., & Cohen, P. (1987). *Applied Multiple Regression/Correlation Analyses for The Behavioral Sciences* (3rd. ed.). Hillsdale, NJ.: Lawrence Erlbaum.

Cooprider, J. G., & Henderson, J. C. (1990). Technology-Process Fit: Perspectives on Achieving Prototyping Effectiveness. *Journal of Management Information Systems, 7*(3), 67-87.

Cossick, K. L., Byrd, T. A., & Zmud, R. W. (1992). A Synthesis of Research on Requirements Analysis and Knowledge Acquisition Techniques. *MIS Quarterly, 16*(1), 117-138.

Daft, R. L., & Lengel, R. H. (1986). Organizational Information Requirements, Media Richness and Structural Design. *Management Science, 32*(5), 554-571.

Driskill, L. P., & Goldstein, J. R. (1986). Uncertainty: Theory and Practice in Organizational Communication. *Journal of Business Communication, 23*(3), 41-56.

Duncan, N. B. (1995). Capturing Flexibility of Information Technology Infrastructure: A Study of Resource Characteristics and their Measure. *Journal of Management Information Systems, 12*(2), 37-57.

Faraj, S., & Sproull, L. (2000). Coordinating Expertise in Software Development Teams. *Management Science, 46*(12), 1554.

Fornell, C., & Larcker, D. F. (1981). Evaluating Structural Equation Models with Unobservable Variables and Measurement Error. *Journal of Marketing Research (JMR), 18*(1), 39-50.

Galbraith, J. R. (1973). *Designing Complex Organizations*. Reading, Mass.: Addison-Wesley Pub. Co.

Gales, L., Porter, P., & Mansour-Cole, D. (1992). Innovation project technology, information processing and performance: A test of the Daft and Lengel conceptualization. *Journal of Engineering and Technology Management, 9*(3-4), 303-338.

Gefen, D., Straub, D. W., & Boudreau, M. (2000). Structural equation modeling and regression: Guidelines for research practice. *Communications of the AIS, 7*, 1-78.

Gerwin, D., & Moffat, L. (1997). Authorizing processes changing team autonomy during new product development. *Journal of Engineering & Technology Management, 14*(3/4), 291.

Ghiselli, E., Campbell, J., & Zedeck, S. (1981). *Measurement theory for the behavioral sciences*. San Francisco: WH Freeman.

Henderson, R. M., & Clark, K. B. (1990). Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms. *Administrative Science Quarterly, 35*(1), 9-30.

Hoegl, M., & Weinkauf, K. (2005). Managing Task Interdependencies in Multi-Team Projects: A Longitudinal Study. *Journal of Management Studies, 42*(6), 1287-1308.

Hoegl, M., Weinkauf, K., & Gemuenden, H. G. (2004). Interteam Coordination, Project Commitment, and Teamwork in Multiteam R&D Projects: A Longitudinal Study. *Organization Science, Organization Science, 15*(1), 38-55.

Hulland, J. (1999). Use of partial least squares (PLS) in strategic management research: a review of four recent studies. *Strategic Management Journal, 20*(2), 195-204.

Jalote, P. (2000). *CMM in Practice: Processes for Executing Software Projects at Infosys*. Boston, MA.: Addison-Wesley

Kazanjian, R. K., Drazin, R., & Glynn, M. A. (2000). Creativity and technological learning: the roles of organization architecture and crisis in large-scale projects. *Journal of Engineering & Technology Management, 17*(3/4), 273.

Khalifa, M., & Cheng, S. (2002). *Adoption of Mobile Commerce: Role of Exposure* Paper presented at the 35th Annual Hawaii International Conference on System Sciences (HICSS'02) Hawaii.

Kraut, R. E., & Streeter, L. (1995). Coordination in Software Development. *Communications of the ACM, 38*(3), 69-81.

Lee, G., & Xia, W. (2005). The ability of information systems development project teams to respond to business and technology changes: a study of flexibility measures. *European Journal of Information Systems, 14*(1), 75.

Limayem, M., Hirt, S. B., & Chin, W. W. (2001, June 27-29). *Intention does ont always matter: The contingent role of habit on IT usage behavior.* Paper presented at the The 9th European Conference on Information Systems, Bled, Slovenia.

Lohmoller, J.-B. (1989). *Latent Variable Path Modeling with Partial Least Squares* Heidelberg: Physica-Verlag.

Marcoulides, G. A., & Saunders, C. (2006). PLS: A Silver Bullet? *MIS Quarterly*, pp. 211-211, from http://ucfproxy.fcla.edu/login?URL=http://search.ebscohost.com/login.aspx?direct=true&db=buh&AN=21145593&site=ehost-live

Mathieson, K., Peacock, E., & Chin, W., W. . (2001). Extending the technology acceptance model: The influence of perceived user resources. *Database for Advances in Information Systems, 32*(3), 86.

McFarlan, F. W. (1989). Portfolio approach to information systems. In *Software risk management* (pp. 17-25): IEEE Press.

Messick, S. (1980). Test validity and the ethics of assessment. *American Psychologist, 35*(11), 1012-1027.

Nidumolu. (1995). The Effect of Coordination and Uncertainty on Software Project Performance: Residual Performance Risk as an Intervening Variable. *Information Systems Research, 6*(3), 191-219.

Nidumolu, S. R. (1996). A comparison of the structural contingency and risk-based perspectives on coordination in software-development projects. *Journal of Management Information Systems, 13*(2), 77.

Olson, E. M., Walker, J. O. C., & Ruekert, R. W. (1995). Organizing for effective new product development: The moderating role of product innovativeness. *Journal of Marketing, 59*(1), 48.

Pellegrinelli, S. (1997). Programme management: organising project-based change. *International Journal of Project Management, 15*(3), 141-149.

Podsakoff, P. M., MacKenzie, S. B., Lee, J.-Y., & Podsakoff, N. P. (2003). Common method biases in behavioral research: A critical review of the literature and recommended remedies. *Journal of Applied Psychology, 88*(5), 879-903.

Ragatz, G. L., Handfield, R. B., & Petersen, K. J. (2002). Benefits associated with supplier integration into new product development under conditions of technology uncertainty. *Journal of Business Research, 55*(5), 389-400.

Sanchez, R., & Mahoney, J. T. (1996). Modularity, Flexibility, and Knowledge Management in Product and Organization Design. *Strategic Management Journal, 17*(Winter Special Issue), 63.

Schmidt, R., Lyytinen, K., Keil, M., & Cule, P. (2001). Identifying Software Project Risks: An International Delphi Study. *Journal of Management Information Systems, 17*(4), 5-36.

SEI. (1994). *The Capability Maturity Model: Guidelines for Improving the Software Process*. Reading, MA: Addison-Wesley.

Smith, H. J., Keil, M., & Depledge, G. (2001). Keeping Mum as the Project Goes Under: Toward an Explanatory Model. *Journal of Management Information Systems, 18*(2), 189-227.

Song, M., & Montoya-Weiss, M. M. (2001). The Effect of Perceived Technological Uncertainty on Japanese New Product Development. *The Academy of Management Journal, 44*(1), 61-80.

Van De Ven, A. H., Delbecq, A. L., & Koenig, R. (1976). Determinants of Coordination Modes Within Organizations. *American Sociological Review, 41*(2), 322-338.

von Hippel, E. (1990). Task partitioning: An innovation process variable. *Research Policy, 19*(5), 407-418.

Waldman, D. A., Ramirez, G. G., House, R. J., & Puranam, P. (2001). Does Leadership Matter?

CEO Leadership Attributes and Profitability under Conditions of Perceived Environmental Uncertainty. *Academy of Management Journal, 44*(1), 134-143.

Weick, K. E., & Roberts, K. H. (1993). Collective Mind in Organizations: Heedful Interrelating on Flight Decks. *Administrative Science Quarterly, 38*(3), 357-381.

Werts, C., Linn, R., & Joreskog, K. (1974). Intraclass Reliability Estimates: Testing Structural Assumptions *Educational and psychological measurement*, *34*(1), 25.

CHPATER FOUR:
GENERAL CONCLUSION

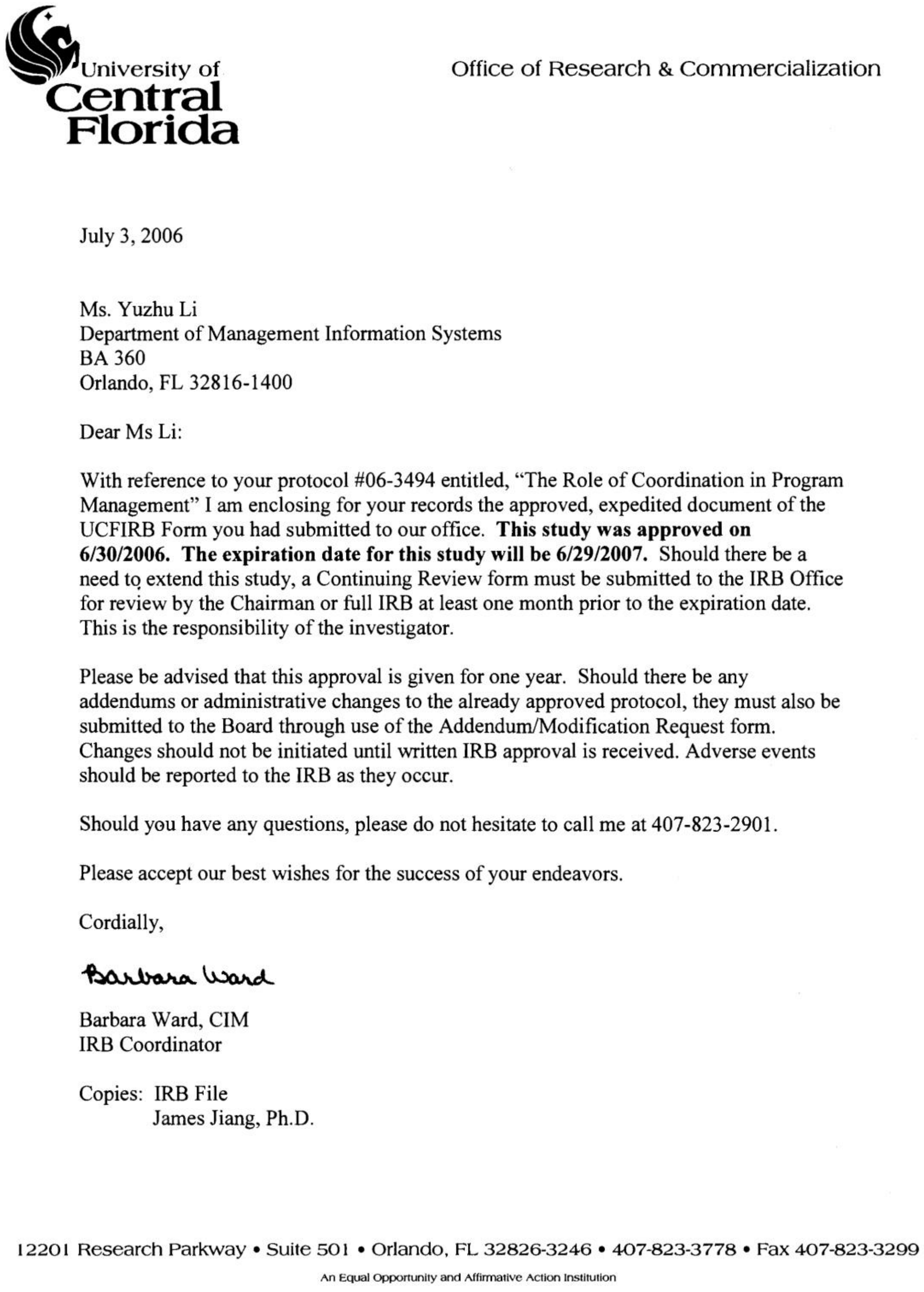
New product development is becoming the nexus of competition as technical and market changes can never be fully controlled. Proactive product development can influence the competitive success, adaptation, and renewal of organizations. Product development programs are started to be deployed in industry because of its effectiveness in resource utilization, execution of product development plans and adaptability to the new changes. This dissertation examines the communication issues of product development programs and contributes to the literature by including product development characteristics in the investigation of communication issues. This dissertation provides problem-solving strategies for product development programs and prescribes the communication strategies for the programs to acquire resources and respond to business and technological changes. This dissertation consisted of two studies focusing on the coordination effectiveness outside and within a software product development program for the characteristics of software product development.

The first study investigated how the product development program deals with the different types of interdependence with the organizational internal environment and the uncertainty from the external environment. Different communication strategies were proposed for each type of interdependence of a software product development with the large organizational context. The empirical results of this study indicated that the communication efforts with the purposes of filtering and molding can moderate the relationship between goal interdependence and program performance. In terms of resource interdependence, the program manager can communicate with other business function heads, convince them the importance of the software development program, lobby for resources and engage them in the feedback loop. When reward interdependence is high, an appropriate level of filtering can ensure the execution of product development plans. Program management teams should build a protective layer for the projects in the program and achieve appropriate extent of isolation by planning and guiding.

The second study closely examined the inter-project coordination in a software development program that consists of multiple inter-related projects. This study argues that administrative and expertise coordination should be used to manage the software development risks in terms of requirement uncertainty and technological uncertainty. The empirical results partially supported that administrative and expertise coordination can be used to manage requirements diversity and technological unpredictability. When requirement diversity is low, a high level of administrative coordination will partition the product development tasks and assign to individual projects with clear goals and evaluation standards. However when requirement uncertainty is high, a high level of administrative coordination will change the assignment of project tasks and generate less impacts on project performance than a low level of administrative coordination. The data analysis also showed that expertise coordination has the moderating effects on requirement diversity. When requirement diversity is low, project teams can recognize when expertise is needed easily and bring the expertise to solve the problems quickly. However when requirement diversity is high, project teams have to spend a lot of time and efforts to figure out the problems and recognize when expertise is needed. It is also difficult to bring the experts to the teams when the new problems cannot be articulated. Technological uncertainty had the fundamental negative impacts on product development. A low level of technological unpredictability can be dealt by a high level of administrative coordination. When a high level of technological unpredictability occurs, individual projects are easily escalated because of changing project tasks and goals. The technological changes will be detrimental and cause the project escalation.

In sum, based upon the Information Processing Theory and Resource Dependence Theory, this dissertation takes a contingency approach to examine the interactions between communication/coordination and product development characteristics including interdependence and uncertainty. This dissertation advances the understandings of program management and provides specific strategies for different contingent conditions. Future research can further examine any other managerial actions that program managers can take to manage the special characteristics of software product development. For example, the teamwork quality of program management teams will be a critical antecedent of the program performance. Future research can examine the factors that will lead to the high teamwork quality of program management teams. In addition, future research can explore other types of programs. Pellegrinelli (2002) observed three different reasons for building programs: maximizing the use of resources , achieving a common goal and integrating with existing processes. Three archetypes of programs are developed for these different reasons: portfolio program, goal-oriented program and heart-beat program. Product development programs belong to the goal-oriented programs. Future research can explore the characteristics of the other types of programs. For example, heart-beat programs are commonly used in executing a strategic business change in the organizations. The implementation of an Enterprise Resource Planning (ERP) system is an example of heart-beat programs. The heart-beat programs usually involve multiple stakeholders. It is very difficult for the diversified stakeholders to agree on evaluation standards of program performances. It will be interesting to explore the communication issues and strategies of heart-beat programs.

APPENDIX: IRB APPROVAL FORM



University of Central Florida

Office of Research & Commercialization

July 3, 2006

Ms. Yuzhu Li
Department of Management Information Systems
BA 360
Orlando, FL 32816-1400

Dear Ms Li:

With reference to your protocol #06-3494 entitled, "The Role of Coordination in Program Management" I am enclosing for your records the approved, expedited document of the UCFIRB Form you had submitted to our office. **This study was approved on 6/30/2006. The expiration date for this study will be 6/29/2007.** Should there be a need to extend this study, a Continuing Review form must be submitted to the IRB Office for review by the Chairman or full IRB at least one month prior to the expiration date. This is the responsibility of the investigator.

Please be advised that this approval is given for one year. Should there be any addendums or administrative changes to the already approved protocol, they must also be submitted to the Board through use of the Addendum/Modification Request form. Changes should not be initiated until written IRB approval is received. Adverse events should be reported to the IRB as they occur.

Should you have any questions, please do not hesitate to call me at 407-823-2901.

Please accept our best wishes for the success of your endeavors.

Cordially,

Barbara Ward

Barbara Ward, CIM
IRB Coordinator

Copies: IRB File
James Jiang, Ph.D.

12201 Research Parkway • Suite 501 • Orlando, FL 32826-3246 • 407-823-3778 • Fax 407-823-3299

An Equal Opportunity and Affirmative Action Institution