

2008

Coordinating Multiple Interdependent Projects in Innovative Product Development Programs

Yuzhu Li

University of Massachusetts at Dartmouth, yuzhu.li@umassd.edu

Ting Lie

Yuan Ze University, tinglie@saturn.yzu.edu.tw

James J. Jiang

University of Central Florida, jjiang@bus.ucf.edu

Gary Klein

University of Colorado at Colorado Springs, gklein@uccs.edu

Follow this and additional works at: <http://aisel.aisnet.org/irwitpm2008>

Recommended Citation

Li, Yuzhu; Lie, Ting; Jiang, James J.; and Klein, Gary, "Coordinating Multiple Interdependent Projects in Innovative Product Development Programs" (2008). *International Research Workshop on IT Project Management 2008*. 7.
<http://aisel.aisnet.org/irwitpm2008/7>

This material is brought to you by the International Research Workshop on IT Project Management (IRWITPM) at AIS Electronic Library (AISeL). It has been accepted for inclusion in International Research Workshop on IT Project Management 2008 by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Coordinating Multiple Interdependent Projects in Innovative Product Development Programs

Yuzhu Li

Decision & Information Sciences Department
University of Massachusetts at Dartmouth
yuzhu.li@umassd.edu

Ting Lie

Department of Information Management
Yuan Ze University
tinglie@saturn.yzu.edu.tw

James J. Jiang

Department of Management Information
Systems
University of Central Florida
jjiang@bus.ucf.edu

Gary Klein

College of Business and Administration
University of Colorado at Colorado Springs
gklein@uccs.edu

ABSTRACT

A software product program is usually knowledge intensive, having highly interdependent projects and a high level of product development uncertainty. Different types of knowledge and expertise are viewed as critical resources that the software development program must acquire and manage. Based upon contingency theory, this study examines the effects of inter-project coordination between multiple project teams within a software product development program. A new model is proposed to examine each type of uncertainty and the moderating effects of inter-project coordination including administrative and expertise coordination. Four hypotheses are developed for empirical testing. Data is under collection from project managers working in IT product development programs. The results will provide an in-depth understanding on how inter-project coordination can effectively reduce the negative impact from requirement and technological uncertainty. The expected results are discussed and future research pointed out.

Keywords

Program management, project management, uncertainty, administrative coordination, expertise coordination

INTRODUCTION

Large scale software development is usually completed in the forms of a collection of projects. 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. This development process is challenging because of different project schedules, a high level of project interdependence and substantial communication difficulties across project boundaries (Brusoni et al., 2001, Gerwin and Moffat, 1997, Hoegl et al., 2004, Kazanjian et al., 2000). The innovativeness and complexity in nature of software product development demands a large amount of coordination and information exchange among project teams. 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 programs, this study focuses on the product development programs that develop complex software systems or the products that are integrated by software. More specifically, this study examines the management issues of a software product development program.

A software product development program is usually managed by a program management team which consists of project managers and program managers (product managers). 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. 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 the projects' activities will achieve 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.

Researchers have viewed business and technology changes as critical software development risks (Boehm, 1991, Schmidt et al., 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 and Weinkauff, 2005, Hoegl et al., 2004), the interaction between the extent of inter-project coordination and uncertainty is still unknown.

In addition, past studies of large scale projects (Hoegl and Weinkauff, 2005, Hoegl et al., 2004) only examined the management of tangible and economic resource dependencies, which is defined as administrative coordination (Faraj and Sproull, 2000). Knowledge is a type of intangible but critical resource which is crucial for non-routine, intellectual team work. 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 (Faraj and Sproull, 2000).

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. Requirements uncertainty and technological uncertainty are considered. 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.

A literature review of uncertainty and coordination will be briefly presented and a theoretical framework proposed based upon Information Processing Theory (Galbraith, 1973). Following the theoretical framework, the hypotheses are developed in the Hypotheses Development section. Planned research methodology follows and future research is addressed at the end.

LITERATURE REVIEW

Software development uncertainty comes from the changes in the business and technological environment (Lee and Xia, 2005). The Information Systems Development Project’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. Business changes are usually signaled in the competitive market first. When business changes are recognized, they are translated into requirements by the program management team. Requirements uncertainty in software development processes has been widely studied because of the difficulty of eliciting requirements from users (Cossick et al., 1992, Nidumolu, 1995). Requirements uncertainty has three dimensions: requirement instability, requirement diversity and requirement unanalyzability (see Table 1 for the definition of each dimension) (Nidumolu, 1996). A high level of requirements uncertainty will need more coordination efforts and lead to less process control and product flexibility (Nidumolu, 1996).

Uncertainties	Dimensions	Definitions
Requirements 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)		

Table 6: Definitions of uncertainty

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 et al., 1996). Typically, a development team deals with two types of information technologies: software development tools and infrastructure technologies (Cooprider and Henderson, 1990). Technology continually creates 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 and Montoya-Weiss, 2001). Perceived technological uncertainty has negative impacts on the costs of new product development and will result in distinct managerial action (Ragatz et al., 2002, Song and Montoya-Weiss, 2001).

A software product program can be more effective in responding to requirement and technological uncertainty if the program team can manage the inter-project coordination well. The software product that a program develops is usually more complex than the product that a single project can handle. Administrative coordination is usually determined by the extent of the existing knowledge and expectations on the development process. However, cross-project administrative coordination cannot be sufficient because of the emerging requirements and technological uncertainties. The innovative nature of software development requires not only the presence of experts in the program but also the patterned interactions and solution-seeking processes for the new problems and task exceptions.

This expertise coordination process has three dimensions: knowing where expertise is located, recognizing where it is needed, and bringing it to bear (Faraj and Sproull, 2000). 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. Expertise is context specific and emerges from patterned interactions and practices (Faraj and 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.

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.

Although the direct effects of uncertainty and coordination on program performance are clearly understood, how and when the inter-project coordination mechanisms reduce the negative impact of uncertainty is not answered yet. 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 1).

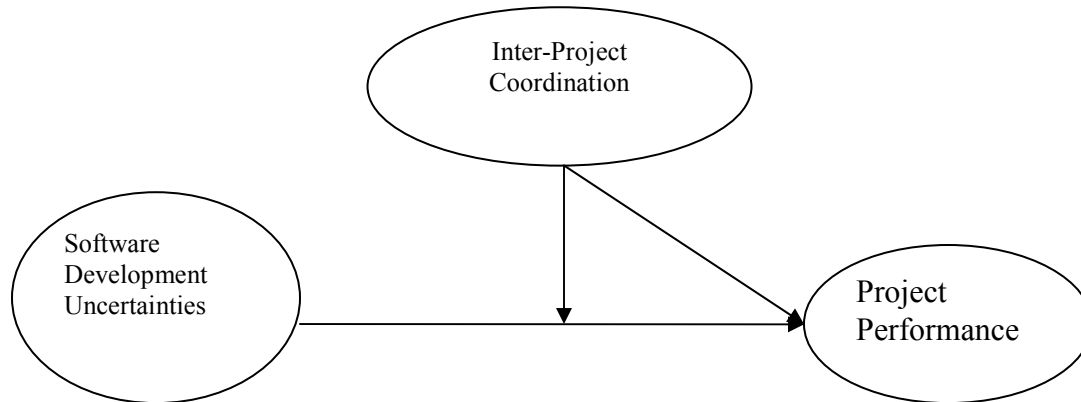


Figure 1. Theoretical Framework of Inter-project Coordination

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.

HYPOTHESES DEVELOPMENT

Requirements uncertainty reflects the changes in the business environment and the interests of different stakeholders. Requirements can be changed in terms of scope and/or the extent of diversified user interests. When the requirements 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 support 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 requirements 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 requirements 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. From this, the following hypothesis is developed.

H1: The magnitude of the relationship between requirements uncertainty and project performance changes as the level of administrative coordination changes.

When requirements uncertainty is high, the estimation of the impact of the possible change is critical for the product development program. 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 requirements 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 through 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: The magnitude of the relationship between requirements uncertainty and project performance changes as the level of inter-project expertise coordination changes.

Technologies are understood as the bodies of knowledge, or understanding and practice, that underpin product design and manufacturing (Brusoni et al., 2001). When technology uncertainty is low, the current program structure and the interfaces between multiple project teams are sufficient to handle communication needs. 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: The magnitude of the relationship between technology uncertainty and project performance changes as the level of administrative coordination changes.

Brusoni et al. (2001) argue that technological changes are more than the component change and involve changes in product architecture. Knowledge integration and application will be critical for the product development success. 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: The magnitude of the relationship between technology uncertainty and project performance changes as the level of inter-project coordination changes.

RESEARCH METHODOLOGY

A survey is used to collect data and test the hypotheses. A pilot study was administered in summer 2006. The survey questions were adjusted based upon the pilot study results. The formal data collection is on-going. The data collection unit is a “program”. On average each program includes 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 sample size is expected to be more than 100. At this point, the number of returned questionnaires is about 70.

The researchers successfully obtained the support from a prestigious university. They are allowed to contact alumni from the Engineering School of this University. Because this university is well known for its strong Software Engineering program, it has a large number of alumni and work in diversified industries. If the alumni agree to participate in the surveys, they will be asked to provide further contacts for the researchers.

Constructs

All the constructs are adopted from the existing literature. However the questions are re-worded according to the software development program context. A group of experts examined the survey questions and did card sorting to assure the validity issues. All the items are measured on a five-point Likert scale, ranging from “to a large extent” (5) to “not at all” (1). The constructs are:

- *Inter-project Administrative Coordination*: The measure for inter-project administrative coordination has six items from Kraut and Streeter (1995). A sample item is “the extent of using formal policies and procedures for coordinating the projects in the program”.
- *Inter-project Expertise Coordination*: The measure for inter-project expertise coordination is 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.
- *Requirements uncertainty*: The measure for requirements uncertainty is adapted from Nidomolu (1995). It has 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 has three items. Requirement analyzability refers to the extent to which a conversion process can be reduced to mechanical steps or objective procedures and has four items.

- *Technology uncertainty*: The measure for technology uncertainty is adapted from Nidumolu (1995). It has two dimensions. Technological unpredictability describes the extent to which unexpected and novel technology occur during the software development product process and has 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 has eight items.
- *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 includes 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 requires the respondents to answer based on the most recently completed projects in the program (1 – Never, 5 – Always) (Nidumolu, 1995).

Hypotheses will be tested and verified by using partial least squares (PLS) analysis. 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. PLS-Graph Version 3.01 will be used to test the hypotheses. The initial data analysis result will be reported at the workshop presentation.

This research will have several limitations. This study only examines a set of moderators. Many moderators that can affect the relationship between uncertainty and project performance will not be examined. The researchers adopt the positivist methodology. The sample size will limit the interpretation of the statistical analysis. The last limitation will be common method bias. Future research can test the validity of the conclusions by using multiple sources of data or a combination of quantitative and qualitative methods.

CONCLUSION

What differentiates this study from previous efforts is 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. It is expected that the results of this research study will highlight the importance of inter-project administrative and expertise coordination in the innovative product development process. Project managers are well trained to execute a project and focus on the projects at hand. However, the increasing interdependence and complexity in software product development demand the project managers to pay attention to the context and be sensitive to the future changes and impacts on their own projects. The right extent of inter-project coordination will help the project managers to 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.

REFERENCES

- Boehm, B., W. (1991) Software Risk Management: Principles and Practices, *IEEE Software*, 8, 32.
- Brusoni, S., Prencipe, A. and 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, 597.
- Burgelman, R. A., Maidique, M. A. and Wheelwright, S. C. (1996) *Strategic management of technology and innovation*, Irwin, Chicago.
- Cooprider, J. G. and Henderson, J. C. (1990) Technology-Process Fit: Perspectives on Achieving Prototyping Effectiveness, *Journal of Management Information Systems*, 7, 67-87.
- Cossick, K. L., Byrd, T. A. and Zmud, R. W. (1992) A Synthesis of Research on Requirements Analysis and Knowledge Acquisition Techniques, *MIS Quarterly*, 16, 117-138.
- Faraj, S. and Sproull, L. (2000) Coordinating Expertise in Software Development Teams, *Management Science*, 46, 1554.
- Galbraith, J. R. (1973) *Designing Complex Organizations*, Addison-Wesley Pub. Co, Reading, Mass.
- Gerwin, D. and Moffat, L. (1997) Authorizing processes changing team autonomy during new product development, *Journal of Engineering & Technology Management*, 14, 291.
- Hoegl, M. and Weinkauff, K. (2005) Managing Task Interdependencies in Multi-Team Projects: A Longitudinal Study., *Journal of Management Studies*, 42, 1287-1308.

- Hoegl, M., Weinkauff, K. and Gemuenden, H. G. (2004) Interteam Coordination, Project Commitment, and Teamwork in Multiteam R&D Projects: A Longitudinal Study, *Organization Science*, 15, 38-55.
- Kazanjian, R. K., Drazin, R. and 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, 273.
- Kraut, R. E. and Streeter, L. (1995) Coordination in Software Development., *Communications of the ACM*, 38, 69-81.
- Lee, G. and 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, 75.
- Nidumolu (1995) The Effect of Coordination and Uncertainty on Software Project Performance: Residual Performance Risk as an Intervening Variable, *Information Systems Research*, 6, 191-219.
- Nidumolu (1996) A comparison of the structural contingency and risk-based perspectives on coordination in software-development projects, *Journal of Management Information Systems*, 13, 77.
- Pellegrinelli, S. (1997) Program management: organizing project-based change, *International Journal of Project Management*, 15, 141-149.
- Ragatz, G. L., Handfield, R. B. and Petersen, K. J. (2002) Benefits associated with supplier integration into new product development under conditions of technology uncertainty, *Journal of Business Research*, 55, 389-400.
- Schmidt, R., Lyytinen, K., Keil, M. and Cule, P. (2001) Identifying Software Project Risks: An International Delphi Study, *Journal of Management Information Systems*, 17, 5-36.
- Song, M. and Montoya-Weiss, M. M. (2001) The Effect of Perceived Technological Uncertainty on Japanese New Product Development, *The Academy of Management Journal*, 44, 61-80.