The factors that affect the implementation success of IS

Shih-Wei Chou
Department of Information Management National Kaohsiung First University of Science of Technology
swchou@ccms.nkfust.edu.tw

Mong-Young He
College of Management National Kaohsiung First University of Science of Technology
mvdis2@kcg.gov.tw

Pi-Yi Chen
College of Management National Kaohsiung First University of Science of Technology
u9328906@ccms.nkfust.edu.tw

Abstract

How to implement an IS (information system) successfully remains a problem. To address this, drawing on information processing theory, this study proposes an integrated model that delineates the relationships among two design variables, risk resolution strategy, and software development success. Experimental design was conducted to test the model. Our results show that risk resolution strategy has a positive effect on software success. As expected, software projects with low goal conflict exhibit better performance of risk resolution strategy, productivity, and process satisfaction than projects with high goal conflict. In addition, organic coordination strategy demonstrates better achievement of risk management, productivity, and process satisfaction than those with mechanistic coordination strategy. Regarding the interaction effect, surprisingly, the findings show the effects on both risk management and process satisfaction are insignificant. Further, as expected the interaction effect on productivity is significant. Implications and discussions are provided.

Key words: project management, goal conflict, coordination strategy, risk management, software development success

1. Introduction

Prior research (Keil 1995, Newman and Sabherwal 1996) has recognized software development as a costly and risky task. Poor project execution may result in quite a few drawbacks such as missed deadline, poor productivity or a software product that does not meet user's requirements. In order to avoid these problems, previous literature proposed two streams of solution. The first stream of tactics has focused on technical innovations involving the development of new and enhanced methods and tools (Fichman and Kemerer 1993, Henderson and Cooprider 1990). Second, a more current stream of tactics entails improved project management practices applied to the processes of software development (Andres and Zmud 2002, Crowston 1997, Hunton and Beeler 1997, Huisman and Livari 2006, Iverson et al. 2004, Shaft and Vessey 2006, Sherif et al. 2006). The above studies concern a variety of issues such as coordination mechanisms, risk management, and work group coordination, as well as the social context such as goal orientation, conflict, and cognitive fit surrounding software developer participation and interaction. This study emphasizes the second stream of tactics.

In terms of IS development process, although prior research has proposed a variety of variables that may influence the success of software, for example management's control over scare resources (Sharma and Yetton 2003), user participation and involvement (Barki et al. 1993), resource interdependence (Larsen 2003), environmental uncertainty and task uncertainty (Andres and Zmud 2002, Karime et al. 2004), and so on, so far there is little work considering IS success from the perspective of information processing. According to information-processing theory, in order to develop an IS (information system) effectively and efficiently, for example high system productivity and high customer satisfaction, system developers have to process information effectively to reduce uncertainty and equivocal information (Karimi et al. 2004, Peffers and Tuunanen 2005). Based on this rationale, this study develops a theoretical model by addressing two aspects that may influence the performance of IS development: IS design variables (Andres and Zmud 2002, Larsen 2003) and risk resolution strategy (Iversen et al. 2004, Tiwana and Keil 2006). Regarding the former, this study selects two salient variables, which may influence the effectiveness of information process or IS success—goal structure and coordination strategy. Since both goal structure and coordination strategy play a key role in influencing the effectiveness of information process, which in turn affects the performance of IS, the first research question addresses the influences of these two factors on IS success. This includes: (1) the individual role of goal structure and coordination strategy; and (2) the fit between these two factors.

The second issue that this study analyzes is risk resolution strategy. Software development, as well as other change or innovative initiatives, are faced with quite a few risks such as requirements volatility, project complexity, lack of management support, lack of interaction between team members, inability to learn from experience, and inability to adjust mission (Iversen et al. 2004, Jiang and Klein 1999, Karimi et al. 2004, Larsen 2003, Tiwana and Keil 2006). There has been little attempt to realize the relative effect of various drivers of risk and how they shape the overall risk, for example, assessing the impact of two different goal structures (cooperatively linked goal v.s. competitively linked goal) on risk resolution strategy. To fill this gap, this study proposes the second research question. This study proposes two risk resolution strategies (adjust mission and modify strategy) according to prior work (Iversen et al. 2004, Ropponen and Lyytinen 2000). While “adjust mission” refers to the modification of the project’s goal to make the assigned task more feasible, acceptable, and useable, “modify strategy” represents adopting new approaches or adapting the initiatives that facilitate the feasibility of the project, such as best practice, improvement strategy and ideas. Then, we analyze the influence of two IS design variables (i.e. goal structure and coordination strategy) on the above resolution strategies, which in turn may affect IS success.

To test the research questions empirically, this study adopted experimental design. The data was collected from 72 subjects in 60 firms. These subjects, belonging to MIS department of a firm, have full time job in the daytime, and have experience of achieving IS project. At the same time, these subjects also enrolled at evening sections of a software engineering course that was offered by MIS department at a public university. We employed both PLS (partial least square) and ANCOVA to examine the research questions and hypotheses. Our unit of analysis was the team.

The primary contribution of this study is simultaneously examining the contingent influence of two design variables on both risk resolution strategy and IS development success. The findings provide some interesting insights into how software projects’ quality is influenced by either design variables or risk resolution strategy. Overall, these contributions bear significant implications for both IS development research and practice.

2. Hypotheses development

2.1 goal conflict

Rahim (1992) defined organizational conflict according to the resources that a software team or an organizational unit uses. Conflict falls into three main categories: substantive conflict (disagreement on task or content issues concerning the solution to a problem), interest conflict (same understanding of a problem bur preference for incompatible solutions), and goal conflict (preferred outcomes or objectives that are inconsistent) (Andres and Zmud 2002, Sheriff et al. 2006). This study emphasizes the goal conflict within software development since it happens often and exerts a significant influence on IS development. Tjosvold (1988) conceptualized organization conflict as two different types of goal interdependence—cooperatively linked or competitively linked. Alper et al’s (1998) empirical study showed that the teams with perceived competitively linked goals have lower productivity and process satisfaction because of the undermining and obstructing the goals of others. This phenomenon can be further explained by other studies (Locke and Latham 1990, Locke et al. 1994), they argued that the motivation for collaboration and synergy decreases when individuals proceed to develop strategies that maximize their own performance only. In contrast, members of a software team with cooperatively linked goal are willing to perform accurate communication, mutual goal adjustments, and task orientation (Alper et al. 1998).

In a software development environment, different stakeholders emphasize dissimilar perspectives of a software project (Robey et al. 1989). Exchanging information among members in an IS team with high goal conflicts is difficult, if not impossible. This is because competing goal not only diminishes motivation of interacting, but also reduces the involvement of contributing precious insights. As a result, both uncertainty and ambiguity are high. This in turn suggests that high goal conflict may lead to lower productivity and lower process satisfaction in terms of information processing theory. Thus, we have the first hypothesis.

Hypothesis 1: IS teams with higher goal conflict will have less IS development success than those with lower goal conflict.

In terms of risk resolution strategy, in order to analyze risks and propose appropriate risk resolution strategy, software development team requires a wide scope of information processing and sharing. This is because risk resolution usually needs information that is timely, has broad scope, has various forms of aggregation, and is integrated (Iveren et al. 2004, Karimi et al. 2004). For a software team with competitively linked goals, the information processing capacity and flexibility to adapt to environmental changes are limited. This result usually leads to poor capability of handling risk resolution. This suggests the second hypothesis.

Hypothesis 2: IS teams with higher goal conflict will have less successful risk resolution strategy than those with lower goal conflict.

2.2 Coordination strategy

Coordination refers to the mode of linking together different parts of an organization to accomplish a set of collective tasks (Van de Ven et al. 1976). According to coordination theory, in order to collaborate and make decisions effectively, a mechanism with communication and control capability is critical.

In an IS development context, as IS developers are usually confronted with a diversity of task requirements, which may be unexpected, constantly changing, difficult to analyze, and interdependent, they need an effective coordination strategy to reduce the uncertainty and ambiguity. This further suggests that the teams using more flexible coordination strategy (organic strategy) may generate higher productivity and process satisfaction than those using less flexible coordination strategy (mechanistic strategy). This leads to the third hypothesis.

Hypothesis 3: IS teams using organic coordination strategy will have more IS development success than those using mechanistic coordination strategy.

In order to achieve risk resolution efficiently and effectively, it is necessary that IS teams use a more flexible coordination strategy (such as organic strategy). The reasons are two-fold. First, the informal and horizontal communication channels provided by organic strategy enable the information sharing in a timely way. In addition, organic strategy facilitates the delegation of decision-making (or decentralized decision making), which indicates the decision time can be reduced because the members who are most sensitive to the problems can make decision independently (Kraut and Streeter 1995, Rasch and Tosi 1992). Due to the timely decision-making and expertise exchange, IS team may adjust its mission and modify strategy in an efficient way. Second, organic strategy also facilitates synchronization of tasks of team members by clarifying both an individual member's task and the interface between them (Karimi et al. 2004). This in turn indicates that team members may adjust mission and modify strategy in an effective way. The above arguments lead to the fourth hypothesis.

Hypothesis 4: IS teams using organic coordination strategy will have more successful risk resolution strategy than those using mechanistic coordination strategy.

2.3 Fit between goal conflict and coordination strategy

Previous studies (Tjosvold 1988, Victor 1990) showed that goal conflict interacts with coordination strategy in determining organizational effectiveness. In other words, these two variables are not independent; the impact of goal conflict on organizational effectiveness (e.g. productivity) is contingent on the coordination strategy. Within software development context, to minimize disputes concerning task responsibilities and allocation of resources, project managers tend to employ mechanistic coordination strategy, including detailed job descriptions, formal chains of command, and specific rules and procedures. Alternatively, researchers (Gersick and Davis-Sacks 1990) also argued that moderate conflict sometimes can increase productivity and team performance. Under a reasonable conflict condition, team members are more likely to challenge the status quo, focus on task, consider the controversial issue from different points of view, and resolve problems in a collective way (Andres and Zmud 2002).

Building on information processing views, the above contingencies of the design variables (i.e. a fit between goal conflict and coordination strategy) that may reduce the uncertainty and ambiguity to a higher extent imply a better performance (Gresov 1989, Van de Ven et al. 1976). Specifically, if members of an IS team have conflicting goals (e.g. while some members of the team emphasize the quality of a software project, others regard timely completion as being more important), mechanistic coordination strategy is a better choice than organic strategy. As a result, this fit situation may induce both higher productivity and more process satisfaction. This leads to the fifth hypothesis.

Hypothesis 5: The extent of the fit between goal conflict and coordination strategy is positively related to IS development success.

The above logic (i.e. fit between goal conflict and coordination strategy) (Gresov 1989, Van de Ven et al. 1976) is applicable to risk resolution strategy. A misfit between goal conflict and coordination strategy (e.g. both "high goal conflict and organic strategy" and "low goal conflict and mechanistic strategy" imply misfit) leads to high uncertainty and ambiguity in terms of information processing. In a risk management context, either adjusting mission or modifying strategy relies on the detailed information regarding the feasibility of IS initiative, improvement strategy, or best practice (Iversen et al. 2004, Tiwana and Keil 2006). Since misfit situations hinder the above activities, risk resolution is difficult to be achieved effectively and efficiently. This suggests the sixth hypothesis.

Hypothesis 6: The extent of the fit between goal conflict and coordination strategy is positively related to the success of risk resolution strategy.

Finally, since risk resolution is aimed at both assessing risks and addressing risks (Du et al. 2007), we may expect a more successful software project provided that this project implements risk resolution strategy successfully. This leads to our final hypothesis.

Hypothesis 7: Risk resolution positively influences software development success.

3. The study's research design

This study used laboratory experiments to test the hypotheses. The experiments were conducted by employing a 2*2 factorial design. The contingency factors employed in this study were two design variables--goal conflict and coordination strategy. While task-process satisfaction and team productivity were adopted to measure the success of software development (i.e. dependent variables), adjust mission and modify strategy (endogenous variables) were used to measure risk resolution strategy. The definitions of these variables are listed in Table 1.

Table 1. Comfirmatory Factor analysis, Constructs, and Item wording

	1	2	3	4	t-value
1.coordination strategy	**1.00**	0.00	0.49	0.63	
2. goal conflict	0.00	**1.00**	0.51	0.58	
3. Risk management					
3.a1 focus on the initiative	0.39	0.45	**0.81**	0.61	7.53***
3.a2 specify the objective	0.42	0.41	**0.81**	0.54	7.18***
3.a3 create results that are conceived to be usable	0.51	0.46	**0.90**	0.60	14.91***
3.a4 use acceptable solutions	0.41	0.33	**0.87**	0.53	13.00***
3.a5 adjust the level of ambition regarding the goal	0.46	0.32	**0.86**	0.54	11.65***
3.b1 document and emphasize best practice	0.48	0.35	**0.89**	0.62	11.62***
3.b2 affect the expectations towards the initiative	0.43	0.35	**0.92**	0.59	16.50***
3.b3 base the initiative on facts and experiences	0.38	0.44	**0.84**	0.60	10.75***
3.b4 create clear and shared visions	0.44	0.52	**0.89**	0.76	20.60***
3.b5 identify and solve specific problems	0.51	0.30	**0.89**	0.60	12.09***
3.b6 use an incremental improvement strategy	0.44	0.43	**0.87**	0.67	12.43***
3.b7 consider alternative improvement ideas	0.47	0.46	**0.89**	0.78	12.46***
3.b8 reuse other's success	0.31	0.58	**0.84**	0.71	6.82***
3.b9 adapt well-known standard solutions	0.43	0.47	**0.81**	0.74	7.57***
3.b10 adapt the strategy to the task	0.12	0.51	**0.78**	0.60	6.06***
3.b11 design effect measures	0.45	0.45	**0.83**	0.75	9.70***
4. Software development success: Concerning the software development process…					
4.c1 the assigned task is fair	0.64	0.45	0.59	**0.83**	13.04***
4.c2 the available resource is fair	0.54	0.54	0.70	**0.92**	18.63***
4.c3 the interaction between members is efficient	0.60	0.55	0.75	**0.92**	24.56***
4.c4 I am satisfied with the results of the IS	0.43	0.48	0.72	**0.86**	10.19***
4.c5 I am satisfied with the time of conducting the IS project	0.42	0.47	0.68	**0.88**	11.27***
Team productivity (function point count)	0.62	0.53	0.48	**0.75**	9.39***

*p<0.1, **p<0.05, ***p<0.01

3.1 Subjects and sub-team composition

Seventy-two subjects were derived from a population of graduate students enrolled in evening sections of a software engineering course that was offered by MIS (management of information systems) department at a university. These students belonged to three sections of a software engineering course that were instructed by the same teacher. The students who participated in this experiment had full-time job in the daytime, and had experience (less than 3 year) of achieving software project or implementing IS as system analysts or application developers in various industries such as banks, software firms, technology firms and manufacturing firms. The students were randomly assigned to the software development groups. Subjects were asked to complete a short software project, which required them to use software development skills such as systems analysis and design, risk management, and implementation (using visual basic (VB) programming language).

Information from a pretest questionnaire that elicited the grade point average, personal ability/experience with VB, and number of systems analysis and design courses was employed. In doing so, we tried to maintain the balance in the assignment of subjects to project teams. A comparison (t-test) of the mean grade point average of the student subjects indicated no significant differences in grade point average.

3.2 Experimental tasks

A simple e-business application was assigned to the subjects; including the implementation of a customer support system. (prototype). For example, the functions contain establishing customer and product database, and order process. The duration of the task for each team did not exceed a total of twelve hours (two three-hour sessions on two consecutive days). Since the task assigned to the project team contains only the fundamentals of software development, a pilot study showed that the task can be accomplished within twelve-hour time period for a three-person team. Task complexity was controlled across team members by ensuring equality in assigned function points for the modules to be completed by team members. Function points were evaluated by assigning weights to and summing the count of outputs from the modules, inputs/outputs of user queries, inputs to the modules, file data items, and data items sent to, shared with, or received from other modules (Freger 1989). The total function point count was 61 across all groups.

4. Experimental manipulations

4.1 Goal conflict

Two conditions were used to investigate the effect of goal conflict situations-- cooperatively linked goal and competitively linked goal. In the former, both team members were assigned the same goal of either product quality or timely completion. For the latter, different goals (product quality versus timely completion) were allocated to each team member; team members were aware that the other team member has divergent goal. In order to implement the product quality, team member was instructed to develop software that contains detailed instructions, has the capability to handle and recovery easily from inaccurate user input, and includes the functions that exceed the minimum requirements. In terms of timely completion, a team member was instructed to complete the task as quickly as possible at the expense of quality whereas he/she implements the minimum functions according to the specifications proposed in the requirement analysis phase.

4.2 Coordination strategies

According to McCann and Galbraith's (1981) theory, coordination strategy contains three key aspects: formality, localization, and cooperativeness. Informal, decentralization, and cooperative strategies are characterized as organic coordination strategies, while the strategies with formal, centralized, and controlling characteristics fall into the category of mechanistic strategy (Boehm and Ross 1989). This study uses formality and cooperativeness dimensions to represent the coordination strategy. The implementation of high formality includes: (1) spatial separation of the team members, (2) limiting information exchange through memos that were transferred through the project leader, and (3) restricting face-to-face communication; instead only formally scheduled meetings are allowed. Low formality was implemented by allowing team members to communicate fact-to-face. The level of cooperativeness depends on the extent to which the team members were able to involve in decision-making activities together as a group during the software development.

So, under the organic coordination situation, team members developed software close to each other and they exchange information and make decision cooperatively. All of the above activities occurred informally at any time. On the other hand, the teams implemented mechanistic coordination strategy followed the high formality principles mentioned in previous paragraph. To implement this in the experiment, two approaches were used. First, besides instructing the team members to restrict their information exchange to the experimental sessions only, assistants collected all materials (such as memo, documents, or manuscript) at the end of each session and returned them to the members when the experiment resumed. Further, two assistants helped to monitor the interaction between members during the breaks in the computer lab. Second, since it is quite infeasible to monitor the members' communication after leaving the computer lab, members were asked to fill out a questionnaire (self-report) to show whether he/she complied with the instructions about mechanistic strategy (i.e. communicating with each other informally was not allowed). The results indicated that nobody violated the above rule.

4.3 Measures of risk resolution strategy and success of software development

While the subjects were given goals and coordination strategy exogenously, the risk resolution strategy was endogenous—determined by the project manager and implemented by all the team members in terms of agile software management. In addition, since agile software management was imposed on the IS team, project managers along with team members performed incremental adaptation. To accomplish this (Pressman 2005), an agile team not only requires customer feedback (so that IS team may achieve the appropriate adaptations), but also delivers software increments (e.g. executable prototypes or a part of an operational system) in short time periods so that adaptation keeps pace with change.

To implement this in the experiment, two approaches were used in the experiment—generating prototypes and conducting risk analyses. First, before delivering the final IS, two prototypes were generated—the first prototype was created before the end of the first day experiment, while the subjects produced the next prototype and the final system on the second day. Second, project managers played dual role—IS users and risk analyzer. For the former, project managers were asked to assume the role of customers (or IS users), and provided comments by using the prototypes; including whether the customers' needs have been satisfied? whether the level of automation concerning a specific function is appropriate? (e.g. do we have to provide the "check order status" function? To what extent can this be achieved?) As a risk analyzer, project managers were responsible for achieving risk resolution—adjust mission and modify strategy. These were done by assessing and using the prototypes. For example, project managers may ask the team members to adjust the level of ambition regarding the goal, or consider alternative improvement ideas. Table 2 illustrates the details (factor analysis and wording) of the risk resolution strategy along with IS success measures. Project managers of those teams with high goal conflict did not override this directive. Instead, these mangers performed risk resolution according to the evaluation of the prototypes. As illustrated in Table 2, the scales items of risk management were evaluated by a five-point (1-5) anchored Likert scale.

To measure IS development success, this study used two dependent variables-- group process satisfaction and team productivity. The former conceptualizes the project success in terms of process performance, whereas the latter implies the product performance of an IS project (Andres and Zmud 2002, Wallace et al. 2004). To assess IS development process satisfaction, this study employed the group process satisfaction scale validated and developed by Green and Taber (1980). Table 2 lists the scale items, which were rated on a five-point (1-5) Likert scale. The reliability (Cronbach's alpha) for the process satisfaction scale was 0.74 (as shown in Table 3).

Table 2. Correlation of constructs, square root of AVE value, composite reliability, Cronbach's Alpha, and AVE

		1	2	3	4	CR	Cronbach's Alpha	AVE
1	Coordination strategy	**1**				1	1	1
2	Goal conflict	0	**1**			1	1	1
3	Risk management	0.49	0.51	**0.86**		0.98	0.97	0.73
4	Process satisfaction	0.63	0.58	0.76	**0.86**	0.95	0.74	0.74

1. The bolded diagonal values are square roots of the AVE (average variance extracted).
2.CR is composite reliability

Table 3. ANCOVA results for risk management

Source of variation	Degree of Freedom	Mean square	F	Significance of F	Power
Coordination strategy	1	218.65	11.23	0.003**	0.89
Goal conflict	1	227.04	11.66	0.003**	0.90
Coordination strategy * Goal conflict	1	37.87	1.94	0.179	0.26
Programming ability	1	2.17	0.75	0.398	0.13

**p<0.01; *p<0.05, Total degree-of-Freedom= 23, R*R= 0.46

4.4. Data analysis and results

In order to validate our research framework, this study employed both PLS (partial least square) and ANCOVA. PLS allows latent constructs to be modeled, and it makes minimal demands in terms of sample size to validate a model compared to alternative structural equation modeling techniques (Chin et al. 2003). Following recommended two-stage analytical procedures, confirmatory factor analysis (see Table 2) was first conduced to evaluate the measurement model; and then, the structural relationships were examined. As shown in Table 3, with an adequate measurement model and an acceptable level of multicollinearity, hypothesis 7 was tested with PLS. As shown in Figure 1, the result is supported as expected. In addition, the results also imply that both coordination strategy and goal conflict affect software development success either directly or indirectly (through risk resolution).

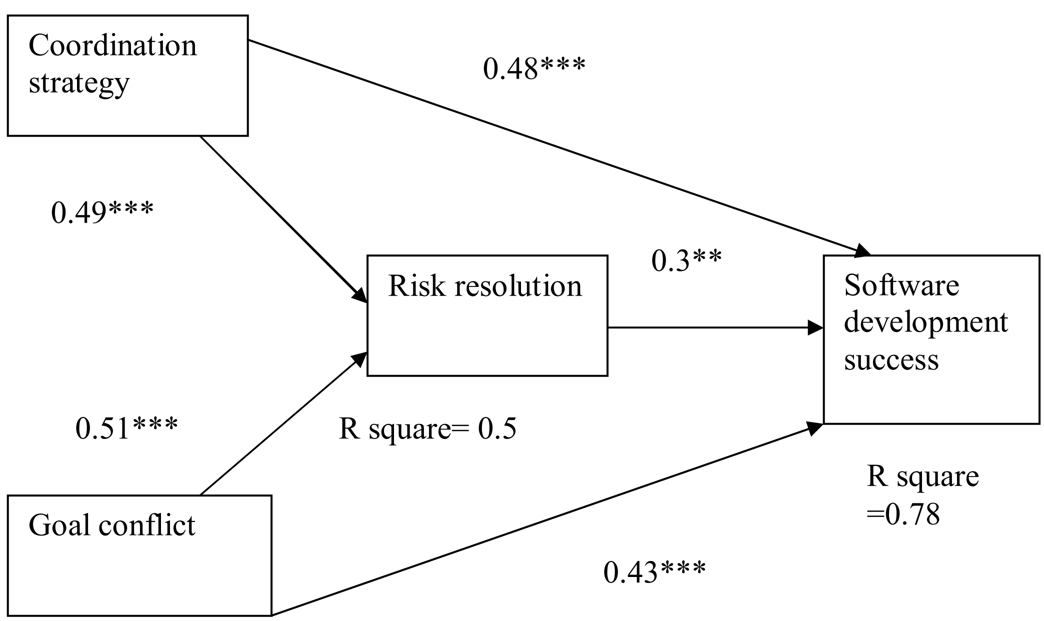

Figure 1. Results of PLS analysis (Note: *p<0.1, **p<0.05, ***p<0.001)

Next, we employed ANCOVA models to examine the impact of IS design variables (or context) on either IS success or risk resolution strategy. Since programming ability may have impact on the productivity, this study chose the former as a covairate while using ANCOVA models to perform statistical analysis. Programming capability was measured by a subject's grade in a C++ programming course taken within a year. As indicated by the biserial and Pearson correlations among the variables, while the correlations between productivity and process satisfaction were not significant, the relations between risk management strategy and IS success (either productivity or process satisfaction) were significant. As expected, the covariate (programming ability) was correlated with productivity. In addition, the confounding effect of this covariate on either treatment is not significant, which indicated that the covariate (programming ability) was completely randomized across all of the experimental conditions.

According to Table 3 to 5, goal conflict exerted a significant influence on risk resolution strategy, satisfaction, and productivity respectively—p values ranged from 0.0001 to 0.001. In addtion, coordination strategy also affected the above variables significantly. Based on the values of both cell means and adjusted marginal means, as expected, three findings were drawn in terms of goal conflict. First, the low goal conflict condition produced more successful risk management than did the high goal conflict condition—while adjusted marginal means (AMM) of low goal conflict was 65.3, AMM of high goal conflict was 59.34. Second, regarding software process satisfaction, low goal conflict situations resulted in higher process satisfaction than did the high goal conflict, because the AMM of the former is larger than that of the latter (18.75> 16.14). Finally, the productivity of the low goal conflict groups is greater than those receiving high goal conflict treatment—11.44> 8.91. Therefore, both hypothesis 1 and 2 were supported.

Table 4. ANCOVA results for software development process satisfaction

Source of variation	Degree of Freedom	Mean square	F	Significance of F	Power
Coordination strategy	1	44.23	17.36	0.001**	0.98
Goal conflict	1	39.53.	15.52	0.001**	0.96
Coordination strategy * Goal conflict	1	0.20	0.08	0.78	0.06
Programming ability	1	0.40	0.16	0.70	0.07
**p<0.01; *p<0.05, Total degree-of-Freedom= 23, R*R= 0.57					

Table 5. ANCOVA results for productivity

Source of variation	Degree of Freedom	Mean square	F	Significance of F	Power
Coordination strategy	1	36.47	36.64	0.0001**	1.00
Goal conflict	1	21.36	21.45	0.0001**	0.99
Coordination strategy * Goal conflict	1	12.76	12.82	0.002**	0.92
Programming ability	1	17.70	17.78	0.0002	0.98
**p<0.01; *p<0.05, Total degree-of-Freedom= 23, R*R= 0.83					

According to the values of both cell means and adjusted marginal means, the impact of coordination strategy can be assessed. We identified three findings that are consistent with the theory. First, the subjects using organic strategy exerted more successful risk strategies than those of mechanistic strategy (AMM= 65.24>59.4). Second, regarding the impact on process satisfaction, organic strategy is still a better choice than mechanistic strategy (18.81 > 16.08). Finally, organic strategy condition has higher productivity since it possessed higher AMM (11.64 > 8.71) than that of mechanistic strategy condition. As a result, our findings supported both hypothesis 3 and 4.

Finally, to address the contingent effect of design variables (i.e. fit between goal conflict and coordination strategy) on either IS success or risk management, we referred to Table 3-5. Surprisingly, the above effect on either risk management or process satisfaction is insignificant. Next, as shown in Table 5, the interaction effect between two design variables was significant (p= 0.002). To further test the relationship between the extent of the fit (between goal conflict and coordination strategy) and productivity, this study employed post hoc pairwise comparisons to assess the relative difference between organic and mechanistic strategy at each level of goal conflict (high and low). As shown in Figure 2, as expected, the above difference (organic strategy is superior to mechanistic strategy) is lower at high goal conflict (mean difference= 1.69) than low goal conflict (mean difference= 4.16). This finding suggested that the productivity is positively related to the extent of the fit between goal conflict and coordination strategy. In sum, while hypothesis 5 is partially supported, hypothesis 6 is not supported.

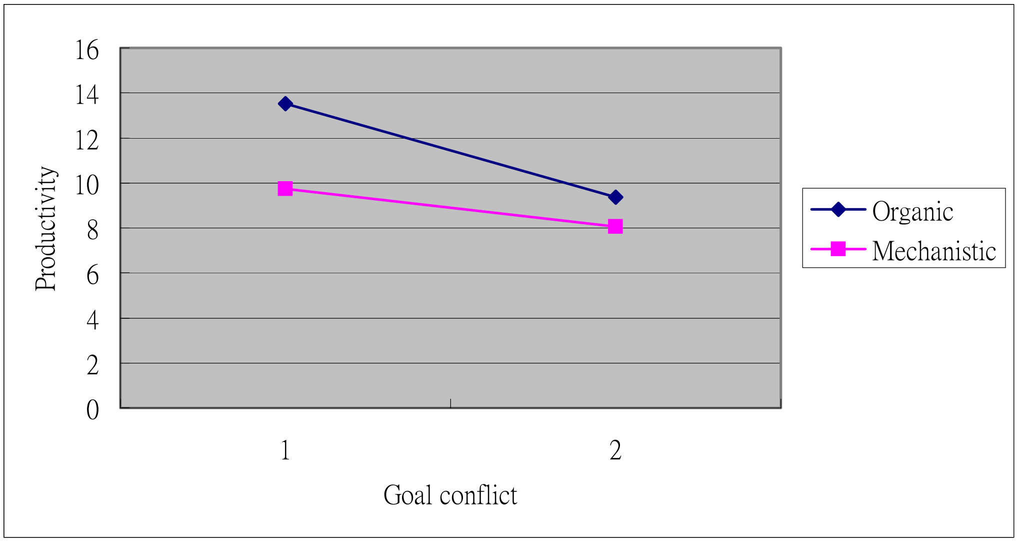

Figure 2 The interaction effect between goal conflict and coordination strategy

5. Discussion and Conclusions

The contributions of this study are two-fold. First, this study proposes a more comprehensive framework concerning IS development than did prior research (Andres and Zmud 2002, Iversen et al. 2004). In our framework, we not only delineate the individual roles of the IS development antecedents (i.e. goal conflict and coordination strategy), but identify the interaction effect of these variables on IS success. In addition, we also address the above effect on risk management which in turn influences the IS success. While most of our findings are consistent with the previous theory, our results only partially support that the proposed fit model—i.e. high goal conflict and mechanism coordination implies an match contingency—will facilitate the IS success and risk management. In sum, project managers acquire insight from the above findings concerning the joint effects as well as the individual impact of the IS design variables on IS success and risk management. Further empirical study is needed to investigate the possible reason(s) why the contingent effect between goal conflict and coordination strategy on IS success is inconsistent with theory.

The second contribution is to specify the role of risk management in influencing IS development. This study implements two kinds of risk strategies—adjust mission and modify strategy. Our results suggest that risk management mediates the relation between IS design variables and IS success. In other words, the goal conflict and coordination strategy influence the performance of risk management, which in turn affects the productivity and process satisfaction. Classical software project management has often relied on the use of predefined methodologies, scheduled periodic design meetings, and formal documents of system design and change activities. Rather than implementing IS in such a static approach, this study suggests that project manager should deal with IS implementation in a more dynamic way. This indicates that, in order to implement a more feasible and effective IS, members should take the software risk into consideration by adjusting mission and modifying strategy from which a more suitable IS functions and implementation method can be produced. Future research may develop a more comprehensive risk strategy by integrating other categories of risk management strategies such as reorganizing the original initiative, and increasing the problem-solving knowledge.

6. References

(1) Alper, S., Tjosvold, D., and Law, K. S. (1998), "Interdependence and Controversy in Group Decision-making: Antecedents of Effective Self-managing Teams," Organizational Behavior and Human Decision Process, 74(1), pp. 33-52.

(2) Amason, A. C. and Schweiger, D. M. (1997), The Effects of Conflict on Strategic Decision-making Effectiveness and Organizational Performance. In C. DeDreu and E.Van de Vliert (eds.), Using Conflict in Organizations. London: Sage, pp. 101-115.

(3) Anderson, N. R. and West, M. A. (1998), "Measuring Climate for Work Group Innovation: Development and Validation of the Team Climate Inventory," Journal of Organizational Behavior, 19(3), pp. 235-258.

(4) Andres, H. P., and Zmud, R. (2002), "A Contingency Approach to Software Project Coordination," Journal of MIS, 18(3), pp. 41-70.

(5) Bhattacherjee, A. and Premkumar, G. (2004), "Understanding Changes in Belief and Attitude toward Information Technology Usage: A Theoretical Model and Longitudinal Test," MIS Quarterly, 28(2), pp. 229-254.
(6) Barki, H., Rivard, S., and Talbot, J. (1993), "Toward an Assessment of Software Development Risk," Journal of MIS, 10(2), pp. 203-225.
(7) Boehm, B. W. and Ross, R. (1989), "Theory-W Software Project Management: Principles and Examples," IEEE Transactions on Software Engineering, 15(7), pp. 902-916.
(8) Buttler, B. S. and Gray, P. H. (2006), "Reliability, Mindfulness, and Information Systems," MIS Quarterly, 30(2), pp. 211-224.
(9) Chin, W. W., Marcolin, B. L., and Newsted, P. R. (2003), "A Partial Lease Squares Latent Variable Modeling Approach for Measuring Interaction Effects: Results from a Monte Carlo Simulation Study and an Electronic-mail Emotion/Adoption Study," Information Systems Research, 14(2), pp. 189-217.
(10) Clark, D. (2000), "Hawthorne effect" (http://www.nwlink.com/~donclark/hrd/history/history.html)
(11) Crowston, K. (1997), "A Coordination Theory Approach to Organizational Process," Organization Science, 8(2), pp. 157-175.
(12) De Dreu, C. K. W. (1997), Productive Conflict: The Importance of Conflict Management and Conflict Issue. In C. De Dreu and E. Van De Vliert (eds.), Using Conflict in Organizations. London: Sage, pp. 9-22.
(13) Delone, W. H., and Mclean, E. R. (2003), "The DeLone and McLean Model of Information Systems Success: A Ten-Year Update," Journal of MIS, 19(4), pp. 9-30.
(14) Dreger, J. B. (1989), Function Point Analysis. Englewood Cliffs, NJ: Prentice Hall.
(15) Du, S., Keil, M., Mathiassen, L., Shen, Y., and Tiwana, A. (2007), "Attention-Shaping Tools, Expertise, and Perceived Control in IT Project Risk Assessment," Decision Support Systems, 43, pp. 269-283.
(16) Fichman, R. G., and Kemerer, C. F. (1993), "Adoption of Software Engineering Software Innovations: The Case of Object Orientation," Sloan Management Review, 34(2), pp. 7-22.
(17) Fornel, C., and Larcker, D. F. (1981), "Structural Equation Models with Unobservable Variables and Measurement Errors," Journal of Marketing Research, 18(2), pp. 39-50.
(18) Gersick, C. J. G., and Davis-Sacks, M. D. (1990), Summary: Task Forces. In J. R. Hackman (ed.), Groups That Work (and Those that Donot): Creating Conditions for Effective Teamwork. San Francisco: Jossey-Bass, pp. 146-153.
(19) Green, S. G. and Taber, T. D. (1980), "The Effects of Three Social Decision Schemas on Decision Group Process," Organizational Behavior and Human Performance, 25(1), pp. 97-106.
(20) Gresov, C. (1989), "Exploring Fit and Misfit with Multiple Contingencies," Administrative Science Quarterly, 34(3), pp. 431-452.
(21) Hair, J. F., Anderson, R. E., and Tatham, R. L., and Black, W. C. (1998), Multivariate Data Analysis (5th ed.), Prentice Hall, Englewood Cliffs, NJ.
(22) Henderson, J. C., and Cooprider, J. G. (1990), "Dimensions of IS Planning and Design Aids: A Functional Model of CASE Technology," Information Systems Research, 1(3), pp. 227-254.
(23) Huck, S. W., Cormier, W. H., and Bounds, W. G. (1974), "Reading Statistics and Research," Harper-Collins, New York.
(24) Huisman, M., and Livari, J. (2006), "Deployment of Systems Development Methodologies: Perceptual Congruence Between IS Managers and Systems Developers," Information and Management, 43(1), pp. 29-49.
(25) Hunton, J. E. and Beeler, J. D. (1997), "Effects of User Participation in Systems Development: A Longitudinal Field Experiment," MIS Quarterly, 21(4), pp. 359-388.
(26) Iversen, J. H., Mathiassen, L., and Nielsen, P. A. (2004), "Managing Risk in Software Process Improvement: An Action Research Approach," MIS Quarterly, 28(3), pp. 395-433.
(27) Jinag, J. J., Klein, G., and Balloun, J. (1998), "Systems Analysis Attitudes Toward Information Systems Development," Information Resource Management Journal, 11(4), pp. 1998.
(28) Jiang, J. J., Klein, G. (1999), "Risks to Different Aspects of Systems Success," Information and Management, 36, pp. 263-272.
(29) Karimi, J., Somers, T. M., and Gupta, Y. P. (2004), "Impact of Environmental Uncertainty and Task Characteristics on User Satisfaction with Data," Information Systems Research, 15(2), pp. 175-193.
(30) Keil, M. (1995), "Pulling the Plug: Software Project Management and the Problem of Escalation," MIS Quarterly, 19(4), pp. 421-447.
(31) Kraut, R. E., and Streeter, L. (1995), "Coordination in Software Development," Communications of the ACM, 38(3), pp. 69-81.
(32) Kwon, T. H. and Zmud, R. (1987), "Unifying the Fragmented Models of Information Systems Implementation," in Critical Issues in Information Systems Research, R. Boland (Ed.), Wiley, Chichester.
(33) Larsen, K. T. (2003), "A Taxonomy of Antecedents of Information Systems Success: Variables Analysis Studies," Journal of MIS, 20(2), pp. 169-246.
(34) Larson, J. R. and Schaumann, L. J. (1993), "Group Goals, Group Coordination, and Group Member Motivation," Human Performance, 6(1), pp. 49-69.
(35) Locke, E. and Latham, G. (1990), A Theory of Goal Setting and Task Performance. Englewood Cliffs, NJ: Prentice Hall.
(36) Locke, E., Smith, K. G., Miriam, E., Chah, D., and Schaffer, A. (1994), "The Effects of Intra-individual Goal Conflict on Performance," Journal of Management, 20(1), pp. 67-81.
(37) Lyytinen, K. and Hirschheim, R. (1987), "Information Systems Failure—A Survey and Classification of the Empirical Literature," in Oxford Surveys in Information Technology, Oxford University Press., Oxford, pp. 257-309.
(38) Lyytinen, K., Mathiassen, L., and Ropponen, J. (1998), "Attention Shaping and Software Risk—A Categorical Analysis of Four Classical Risk Management Approaches," Information System Research, 9(3), pp. 233-255.
(39) McCann, J. E. and Galbraith, J. R. (1981), Interdepartmental Relations. In P. Nystrom and W. Starbuck (eds.), Handbook of Organizational Design, vol. 2, New York: Oxford University Press, pp. 60-84.
(40) Mitchell, T. R. and Silver, W. S. (1990), "Individual and Group Goals When Workers Are Interdependent: Effects on

Task Strategies and Performance," Journal of Applied Psychology, 75(2), pp. 185-193.
(41) Newman, M., and Sabherwal, R. (1996), "Determinants of Commitment to Information Systems Development: A Longitudinal Investigation," MIS Quarterly, 20(1), pp. 23-54.
(42) Olague, H. M., Etzkorn, L. H., Gholston, S., and Quattlebaum, S. (2007), "Empirical Validation of Three Software Metrics Suites To Predict Fault-Proneness of Object-Oriented Classes Developed Using Highly Iterative of Agile Software Development Processes," IEEE Transactions on Software Engineering, 33(6), pp. 402-419.
(43) Peffers, K., and Tuunanen, T. (2005), "Planning for IS Application: A Practical, Information Theoretical Method and Case Study in Mobile Financial Services," Information and Management, 42(3), pp. 483-501.
(44) Pressmen, R. S. (2005), Software Engineering: A Practitioner's Approach, McGraw Hill.
(45) Priem, R. L., Harrison, D. A., and Muir, N. K. (1995), "Structured Conflict and Consensus Outcomes in Group Decision Making," Journal of Management, 21(4), pp. 691-710.
(46) Rahim, M. A. (1992), Managing Conflict in Organizations. New York: Praeger.
(47) Rasch, R. and Tosi, H. (1992), "Factors Affecting Software Developers' Performance: An Integrated Approach," MIS Quarterly, 16(3), pp. 395-413.
(48) Robey, D., Farrow, D. L., and Franz, C. R. (1989), "Group Process and Conflict in System Development," Management Science, 35(10), pp. 1172-1191.
(49) Ropponen, J. and Lyytinen, K. (2000), "Components of Software Development Risk: How to Address Them? A Project Manager Survey," IEEE Transaction on Software Development, 26(2), pp. 98-112.
(50) Schmidt, S., Lyytinen, K., Keil, M., and Cule, P. (2001), "Identifying Software Project Risks: An International Delphi Study," Journal of MIS, 14(4), pp. 5-36.
(51) Shaft, T. M., and Vessey, I. (2006), "The Role of Cognitive Fit in the Relationship between Software Comprehension and Modification," MIS Quarterly, 30(1), pp. 29-55.
(52) Sharma, R., and Yetton, P. (2003), "The Contingent Effects of Management Support and Task Interdependence on Successful Information Systems Implementation," MIS Quarterly, 27(4), pp. 5333-555.
(53) Shenhar, A. J. (1998), "From Theory to Practice: Toward a Typology of Project Management Styles," IEEE Transactions on Engineering Management, 45(1), pp. 33-48.
(54) Sherif, K., Zmud, R. W., and Browne, G. J. (2006), "Managing Peer-to-peer Conflicts in Disruptive Information Technology Innovations: The Case of Software Reuse," MIS Quarterly, 30(2), pp. 339-356.
(55) Snow, A. P., Keil, M., and Wallace, L. (2007), "The Effects of Optimistic and Pessimistic Biasing on Software Project Status Reporting," Information and Management, 44, pp. 130-141.
(56) Tiwana, A. and Keil, M. (2006), "Functionality Risk in Information Systems Development: An Empirical Investigation," IEEE Transaction on Engineering Management, 53(3), pp.412-425.
(57) Tjosvold, D. (1988), "Cooperative and Competitive Dynamics within and between Organizational Units," Human Relations, 41(6), pp.425-436.
(58) Van de Ven, A. H., Delbecq, A. L., and Koenig, R. (1976), "Determinants of Coordination Modes within Organizations," American Sociological Review, 41(2), pp. 322-338.
(59) Victor, B. (1990), Coordinating Work in Complex Organizations, Journal of Organizational Behavior, 11(3), pp. 187-199.
(60) Wallace, L., Keil, M., and Rai, A. (2004), "How Software Project Risk Affects Project Performance: An Investigation of the Dimensions of Risk and an Exploratory Model," Decision Sciences, 35(2), pp. 289-321.